Microprocessor

Third year Comm.

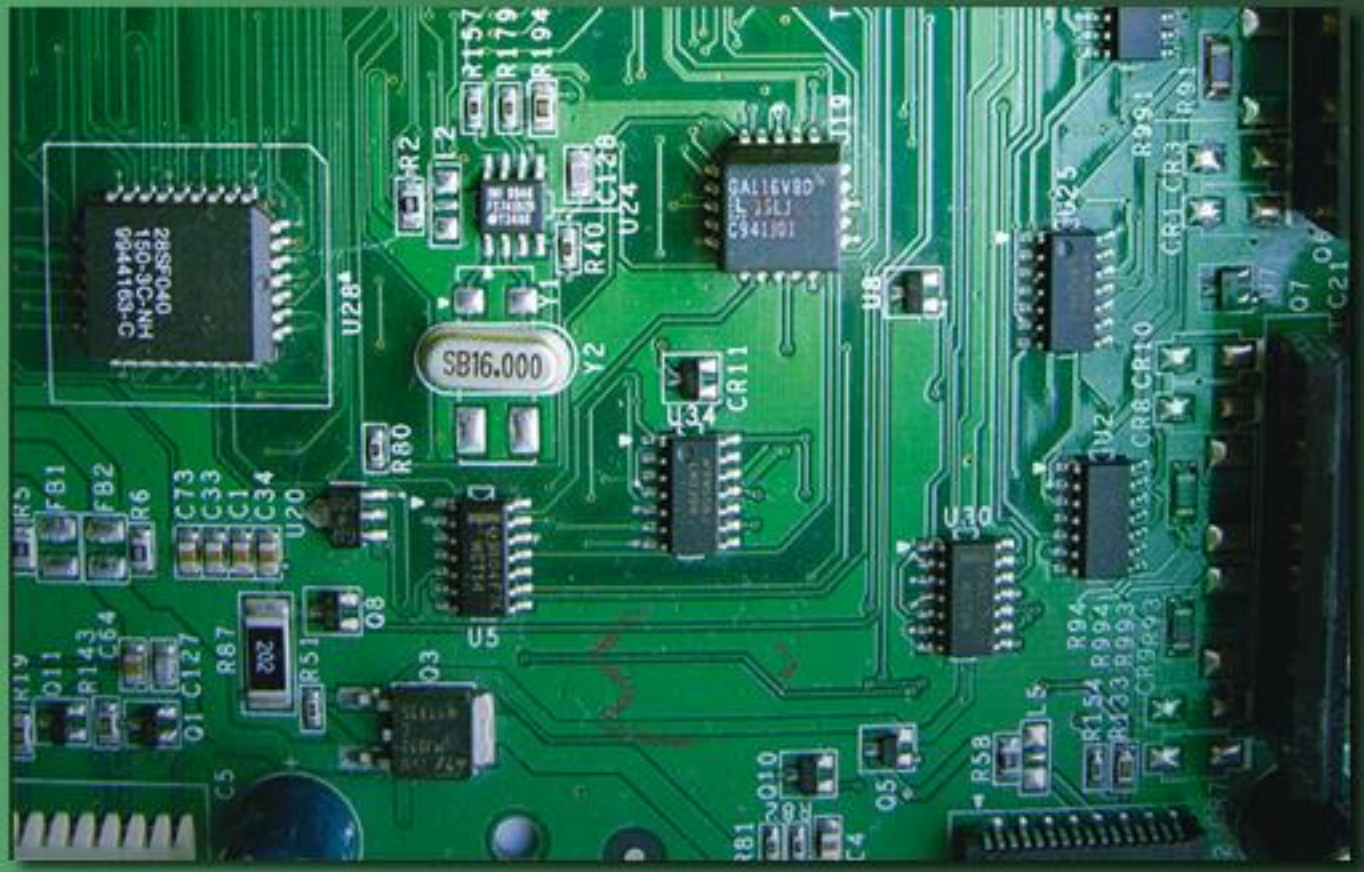
Lect.1 & Lect.2

Dr. Roayat Ismail

The Intel Microprocessors

8086/8088, 80186/80188, 80286, 80386, 80486 Pentium, Pentium Pro Processor, Pentium II, Pentium 4, and Core2 with 64-bit Extensions

Architecture, Programming, and Interfacing



EIGHTH EDITION

Barry B. Brey

PEARSON

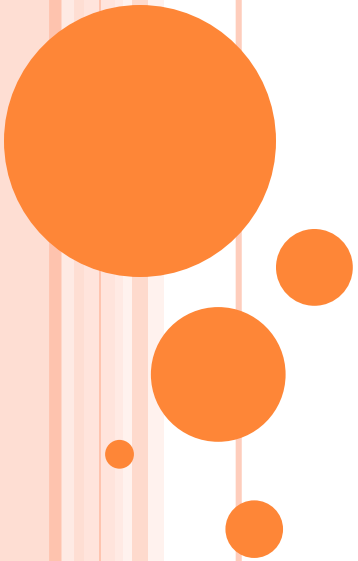
Chapter 2: The Microprocessor and its Architecture

Outlines

- This chapter presents:
 - Introduction to microprocessor and its applications.
 - Evolution of microprocessor
 - The architecture of Intel microprocessors.
 - How its memory space is addressed.
 - Addressing modes are described for the real, protected, and flat modes of operation.
 - Microprocessor internal programming model.

Copyright ©2009 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458 • All rights reserved.

1- Introduction to microprocessor and its applications



Microprocessor Applications

- **Microprocessors are applicable to a wide range of information processing tasks, ranging from general computing to real-time monitoring systems.**
- **The microprocessor facilitates new ways of communication and how to make use of the vast information available at home and in workplace.**
- **Most electronic devices--including everything from computers, remote controls, washing machines, microwaves and cell phones to iPods and more--contain a built-in microprocessor.**
- **Microprocessors are at the core of personal computers, laptops, mobile phones and complex military and space systems.**

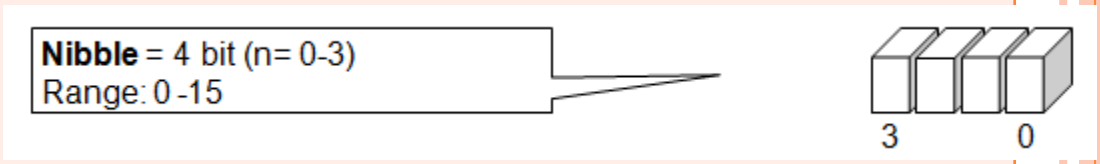
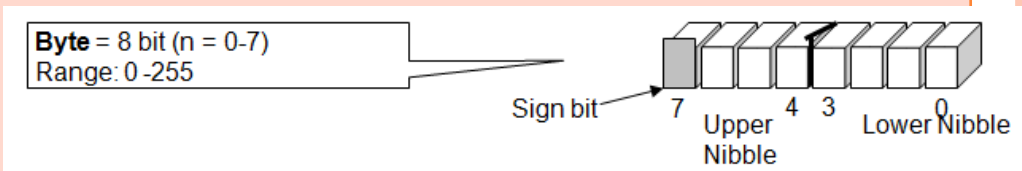
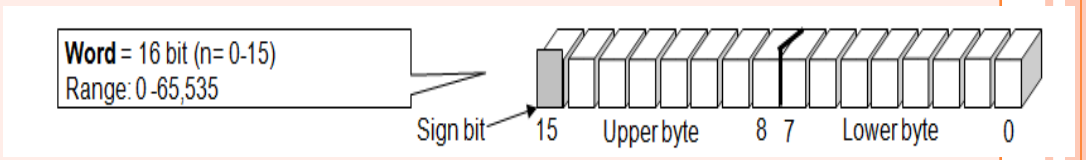
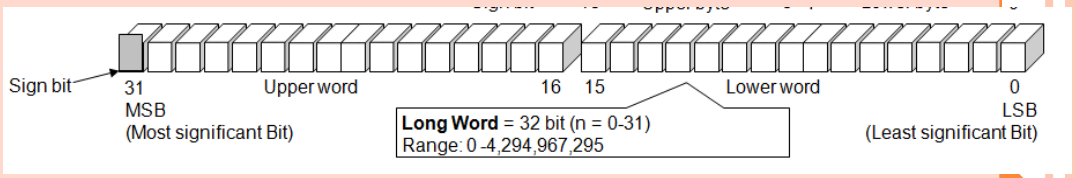
- A microprocessor is usually a silicon chip that contains millions of transistors and other components that process millions of instructions per second integrated with memory chips and other special purpose chips, and directed by software.
- It is a multipurpose, programmable microchip that uses digital data as input and provides results as an output once it processes the input according to instructions stored in its memory.
- They are designed to perform arithmetic and logic operations that make use of data on the chip.
- General purpose microprocessors in PCs are used for multimedia display, computation, text editing and communication.

- Several microprocessors are part of embedded systems. These embedded microprocessors provide digital control to several objects including mobile phones and industrial process control.

- It performs all the computational tasks, calculations and data processing of the computer.
- The most popular type of microprocessor is the **Intel Pentium chip.**

- **What "4 bit/8 bit/16 bit/32 bit microprocessor"?**
- Number of bits that can be processed a time.

DATA SIZE

Nibble	**4 bit**	
Byte	**8 bit**	
Word	**16 bit**	
Long word	**32 bit**	

2. Evolution of microprocessor

- The development of the first microprocessor began in 1969, when Intel engineer Marcian Edward "Ted" Hoff proposed to use a single-chip, general-purpose CPU to perform most computer programming functions. The result was the first microprocessor, the 4004, which was announced by Intel in 1971. This microprocessor evolved into a series of increasingly powerful Intel chips--the 286, 386, 486, and in 1993, the Pentium--for the International Business Machines (IBM) Corp. personal computer (PC) and IBM-compatible PCs.
- Meanwhile, Motorola Corp. developed the 68000 series of chips for the Macintosh personal computer made by Apple Computer. Further advancements have led to robust microprocessors, such as the Intel core i7 microprocessor that is capable of rendering 3D images.

Microprocessor

Fifth Generation **Pentium**

Fourth Generation
During 1980s
Low power version of HMOS technology (HCMOS)
32 bit processors
Physical memory space 2^{24} bytes = 16 Mb

Intel 80386

Third Generation
During 1978
HMOS technology ⇒ Faster speed, Higher packing density
16 bit processors ⇒ 40/ 48/ 64 pins
Easier to program
Physical memory space 2^{20} bytes = 1 Mb

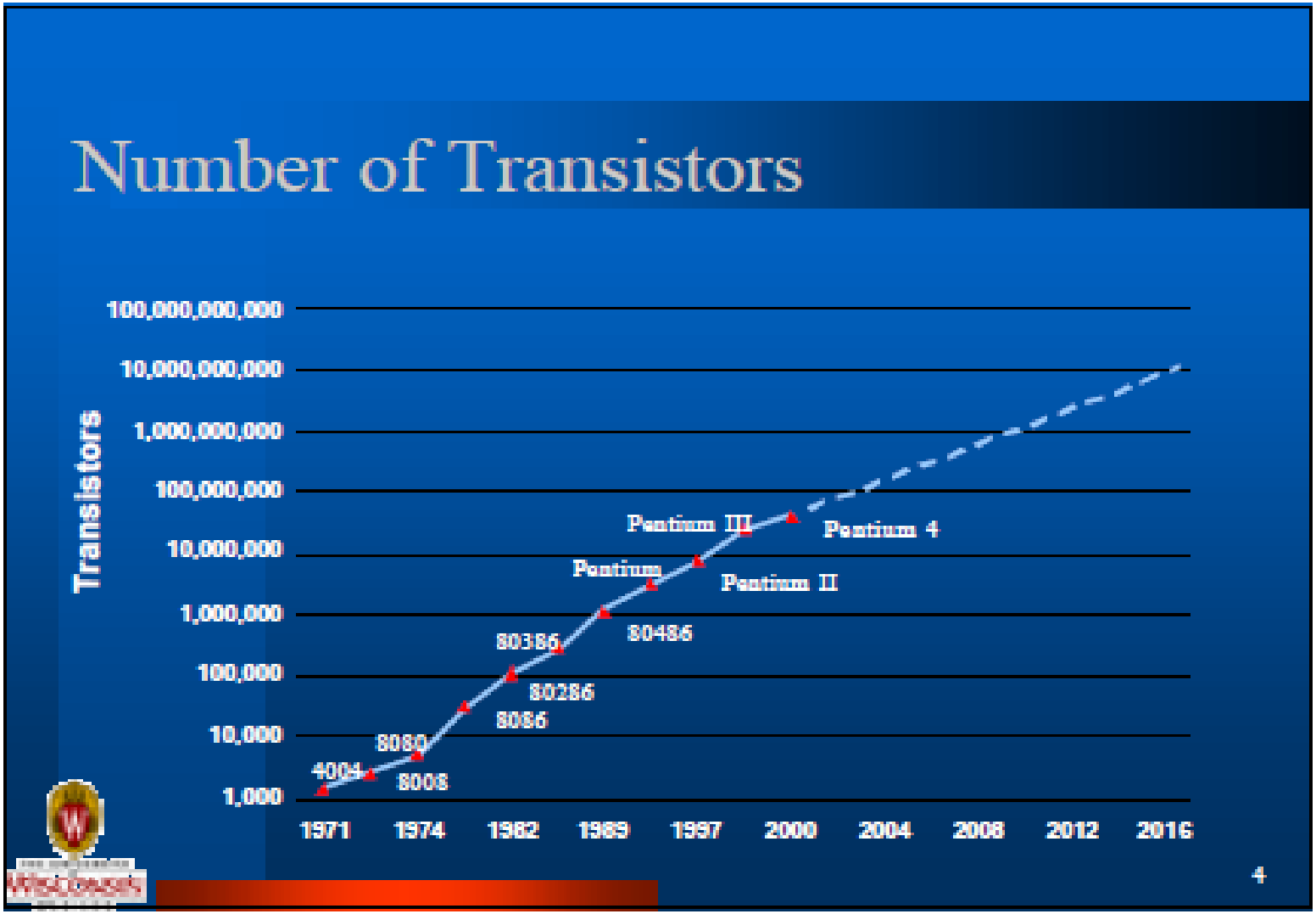
Intel 8086 (16 bit processor)

Second Generation
During 1973
NMOS technology ⇒ Faster speed, Higher density.
4 / 8/ 16 bit processors ⇒ 40 pins
Ability to address large memory spaces and I/O ports

Intel 8085 (8 bit processor)

First Generation
Between 1971 – 1973
PMOS technology
4 bit processors ⇒ 16 pins
8 and 16 bit processors ⇒ 40 pins
Due to limitations of pins, signals are multiplexed

Evolution of Microprocessor



- There are two primary manufacturers of computer microprocessors. Intel and Advanced Micro Devices (AMD) lead the market in terms of speed and quality.
- The impact of microprocessor in different fields is significant. The availability of low cost, low power and small weight, computing capability makes it useful in different applications
- The brain of the computer is microprocessor-drives technology. They are used in computer ranging from microcomputers to supercomputers. In addition, many electronic devices have central processing units (CPU) embedded. The CPU performs computer processing tasks by executing software instructions relative to the data it contains executing complex tasks and instructions.

THE BASIC FUNCTION OF MICROPROCESSOR

- The function of the microprocessor is best described in a three-step process –fetching, decoding and processing (executing).
- In the fetching step: it gets an instruction from the computer's memory.
- In the decoding step, it decides what the instruction means.
- The last step is the processing itself, which involves the microprocessor's carrying out or performing the decoded set of instructions. A modern microprocessor can complete this three-step process millions of times in one second.

3- The architecture of Intel microprocessors.

- Block diagram of a computer system
 - Basic components of a computer system using block diagrams:
 - Cpu
 - Memory
 - Input and output unit
- Internal structure and basic operation of a microprocessor (arithmetic and logic unit, control unit, register sets, accumulator, condition code register, program counter, stack pointer)
- Bus system: data bus, address bus and control bus, connects all the internal components of a computer, such as CPU and memory, to the motherboard.
- Microprocessor clock system
- Examples of microprocessor: 8085,8086.

1.1 DIAGRAM OF A COMPUTER SYSTEM

A **computer** is a programmable machine that receives input, stores and manipulates data//information, and provides output in a useful format.

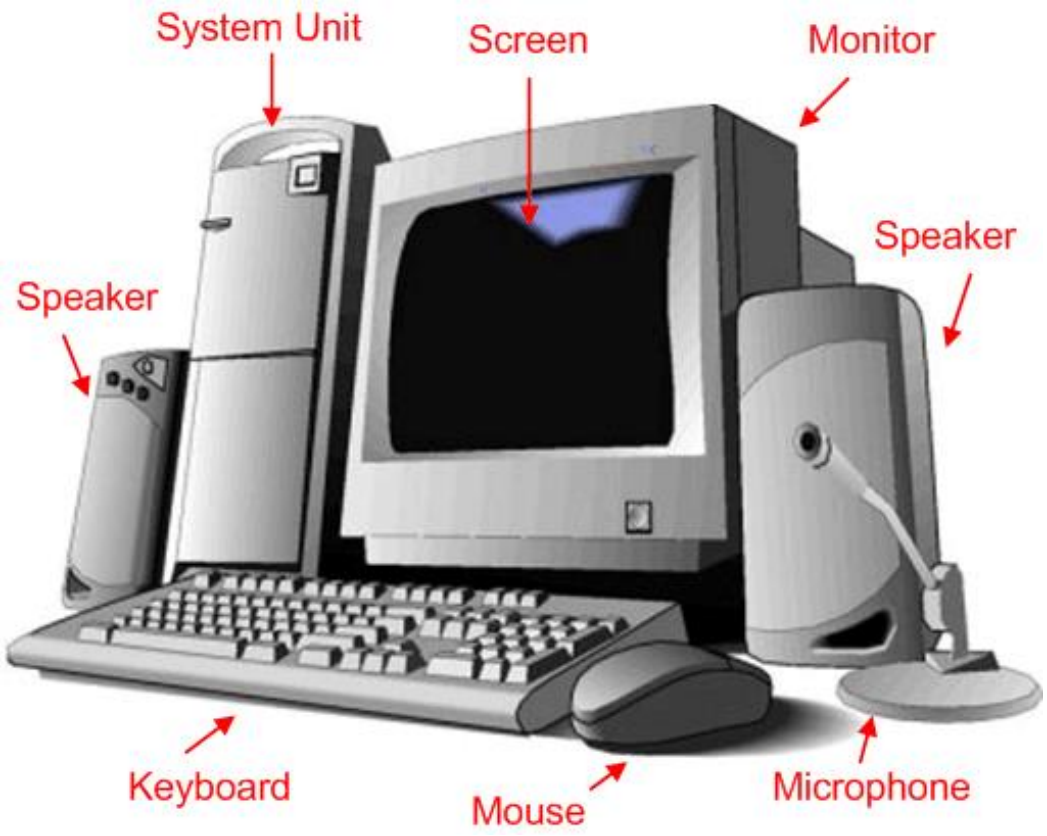
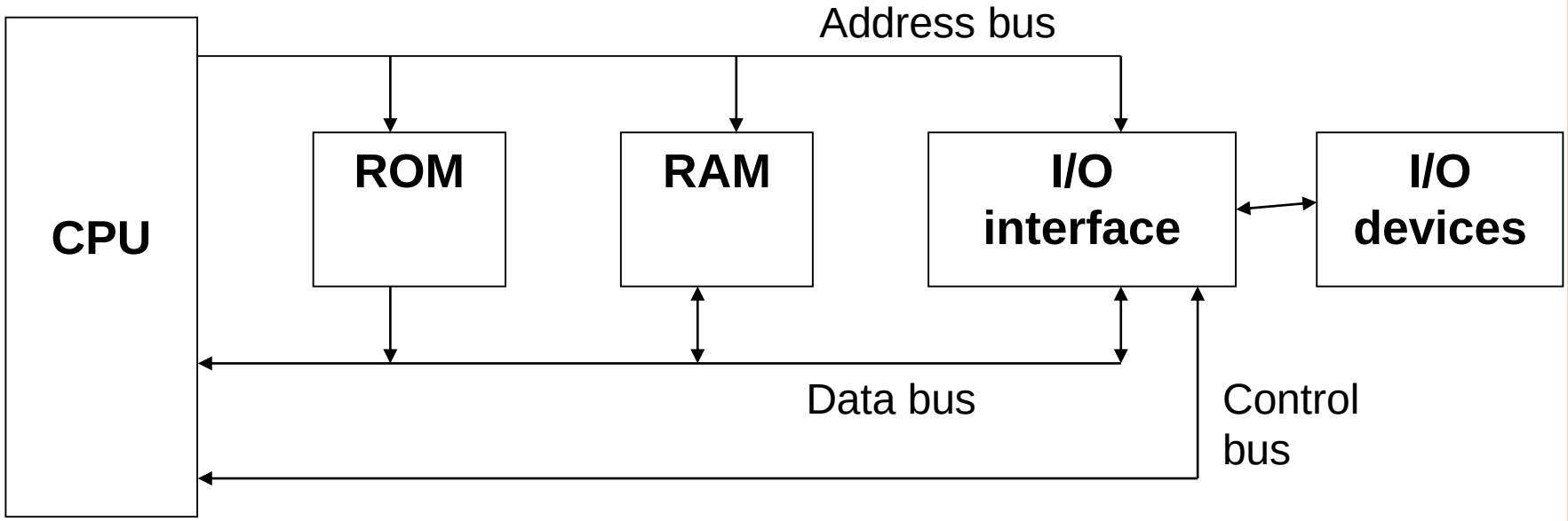


Diagram Of A Computer System

1.1 BLOCK DIAGRAM OF A BASIC COMPUTER SYSTEM

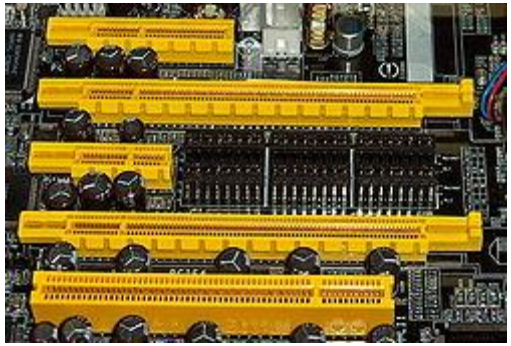
Basic computer system consist of a Central processing unit (CPU), memory (RAM and ROM), input/output (I/O) unit.



Block diagram of a basic computer system

Bus system

- a subsystem that transfers data between computer components inside a computer or between computers.



4 PCI Express bus card slots (from top to bottom: x4, x16, x1 and x16), compared to a traditional 32-bit PCI bus card slot (very bottom).

Data bus

- The data bus is 'bi-directional'
 - data or instruction codes from memory or input/output.are transferred into the microprocessor
 - the result of an operation or computation is sent out from the microprocessor to the memory or input/output.
- Depending on the particular microprocessor, the data bus can handle 8 bit or 16 bit data.

Address bus

- The address bus is '**unidirectional**', over which the microprocessor sends an address code to the memory or input/output.
- The size (width) of the address bus is specified by the number of bits it can handle.
- The more bits there are in the address bus, the more memory locations a microprocessor can access.
- A 16 bit address bus is capable of addressing 65,536 (64K) addresses.

Control bus

- The control bus is used by the microprocessor to send out or receive timing and control signals in order to coordinate and regulate its operation and to communicate with other devices, i.e. memory or input/output.

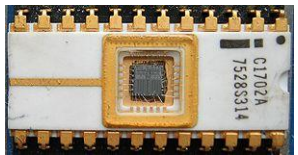
1. CPU - Central Processing Unit
 - the portion of a computer system that carries out the instructions of a computer program
 - the primary element carrying out the computer's functions. It is the unit that reads and executes program instructions.
 - The data in the instruction tells the processor what to do.



Pentium processor

2. Memory

- physical devices used to store data or programs (sequences of instructions) on a temporary or permanent basis for use in an electronic digital computer.
- Computer main memory comes in two principal varieties: random-access memory (RAM) and read-only memory (ROM).
- RAM can be read and written to anytime the CPU commands it, but ROM is pre-loaded with data and software that never changes, so the CPU can only read from it.
- ROM is typically used to store the computer's initial start-up instructions.
- In general, the contents of RAM are erased when the power to the computer is turned off, but ROM retains its data indefinitely.
- In a PC, the ROM contains a specialized program called the BIOS that orchestrates loading the computer's operating system from the hard disk drive into RAM whenever the computer is turned on or reset.



3. I/O Unit

- **Input/output (I/O)**, refers to the communication between an information processing system (such as a computer), and the outside world possibly a human, or another information processing system.
- Inputs are the signals or data received by the system, and outputs are the signals or data sent from it
- Devices that provide input or output to the computer are called peripherals
- On a typical personal computer, peripherals include input devices like the keyboard and mouse, and output devices such as the display and printer. Hard disk drives, floppy disk drives serve as both input and output devices. Computer networking is another form of I/O.

Microprocessor

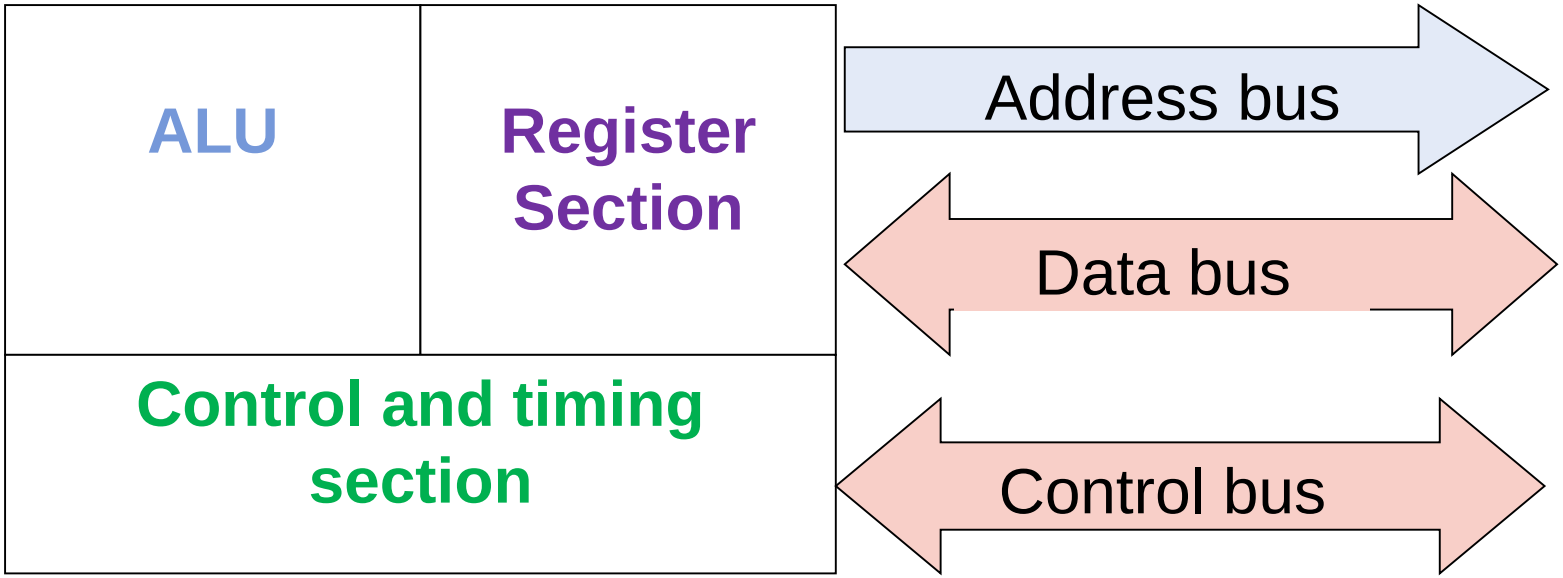
Program controlled semiconductor device (IC) which fetches (from memory), decodes and executes instructions.

It is used as CPU (Central Processing Unit) in computers.

8086 Microprocessor Overview

- **First 16- bit processor released by INTEL in the year 1978**
- **Originally HMOS.**
- **Approximately 29, 000 transistors, 40 pin , 5V supply**
- **Does not have internal clock; external asymmetric clock source with 33% duty cycle**
- **20-bit address to access memory ⇒ can address up to 2^{20} = 1 megabytes of memory space.**

Internal structure and basic operation of microprocessor



Block diagram of a microprocessor

Arithmetic and logic unit (ALU)

- The component that performs the arithmetic and logical operations
- the most important components in a microprocessor, and is typically the part of the processor that is designed first.
- able to perform the basic logical operations (AND, OR), including the addition operation.
- The inclusion of inverters on the inputs enables the same ALU hardware to perform the subtraction operation (adding an inverted operand), and the operations NAND and NOR.

Control unit

- The circuitry that controls the flow of information through the processor, and coordinates the activities of the other units within it.
- In a way, it is the "brain within the brain", as it controls what happens inside the processor, which in turn controls the rest of the PC.
- On a regular processor, the control unit performs the tasks of fetching, decoding, managing execution and then storing results.

Register sets

- The register section/array consists completely of circuitry used to **temporarily store data or program codes until they** are sent to the ALU or to the control section or to memory.

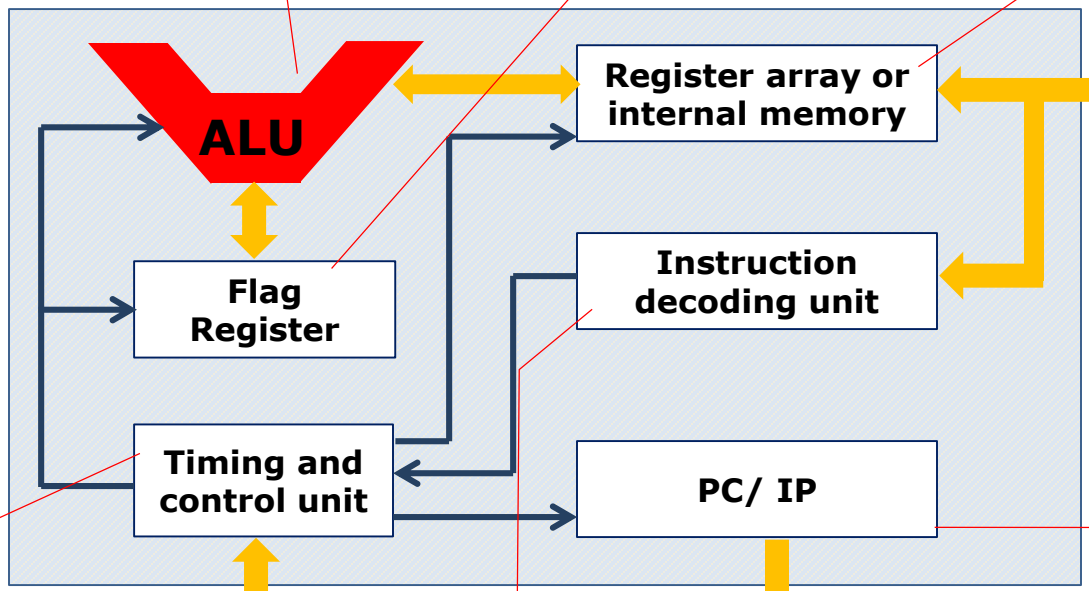
- The number of registers are different for any particular CPU and the more register a CPU have will result in easier programming tasks.

- Registers are normally measured by the number of bits they can hold, for example, an "8-bit register" or a "32-bit register".

Micro processor clock

- Also called clock rate, the speed at which a microprocessor executes instructions. Every computer contains an internal clock that regulates the rate at which instructions are executed and synchronizes all the various computer components.
- The CPU requires a fixed number of clock ticks (or clock cycles) to execute each instruction. The faster the clock, the more instructions the CPU can execute per second. Clock speeds are expressed in megahertz (MHz) or gigahertz ((GHz).
- Some microprocessors are superscalar, which means that they can execute more than one instruction per clock cycle.
- Like CPUs, expansion buses also have clock speeds. Ideally, the CPU clock speed and the bus clock speed should be the same so that neither component slows down the other. In practice, the bus clock speed is often slower than the CPU clock speed, which creates a bottleneck. This is why new local buses, such as AGP, have been developed.

Functional blocks



Computational Unit; performs arithmetic and logic operations

Various conditions of the results are stored as status bits called flags in flag register

Internal storage of data

Generates the address of the instructions to be fetched from the memory and send through address bus to the memory

Generates control signals for internal and external operations of the microprocessor

Decodes instructions; sends information to the timing and control unit

Fetching & Execution Cycles

- **Fetching Cycles**
 - The fetch cycle takes the instruction required from memory, stores it in the instruction register, and
 - moves the program counter on one so that it points to the next instruction.
- **Execute cycle**
 - The actual actions which occur during the execute cycle of an instruction.
 - depend on both the instruction itself and the addressing mode specified to be used to access the data that may be required.

- An **instruction cycle** (sometimes called a **fetch–decode–execute cycle**) is the basic operational process of a computer. It is the process by which a computer retrieves a program instruction from its memory, determines what actions the instruction dictates, and carries out those actions. This cycle is repeated continuously by a computer's central processing unit (CPU), from boot-up to when the computer is shut down.
- In simpler CPUs the instruction cycle is executed sequentially, each instruction being processed before the next one is started. In most modern CPUs the instruction cycles are instead executed concurrently, and often in parallel, through an instruction pipeline:

Components

- 1. The **program counter** (**PC**), commonly called the **instruction pointer** (**IP**) in Intel x86, and sometimes called the **instruction address register.**

It is an incrementing counter that keeps track of the memory address of the instruction that is to be executed next or in other words, holds the address of the instruction to be executed next.

- 2. **Memory address register** **(MAR):**

Holds the address of a block of memory for reading from or writing to.

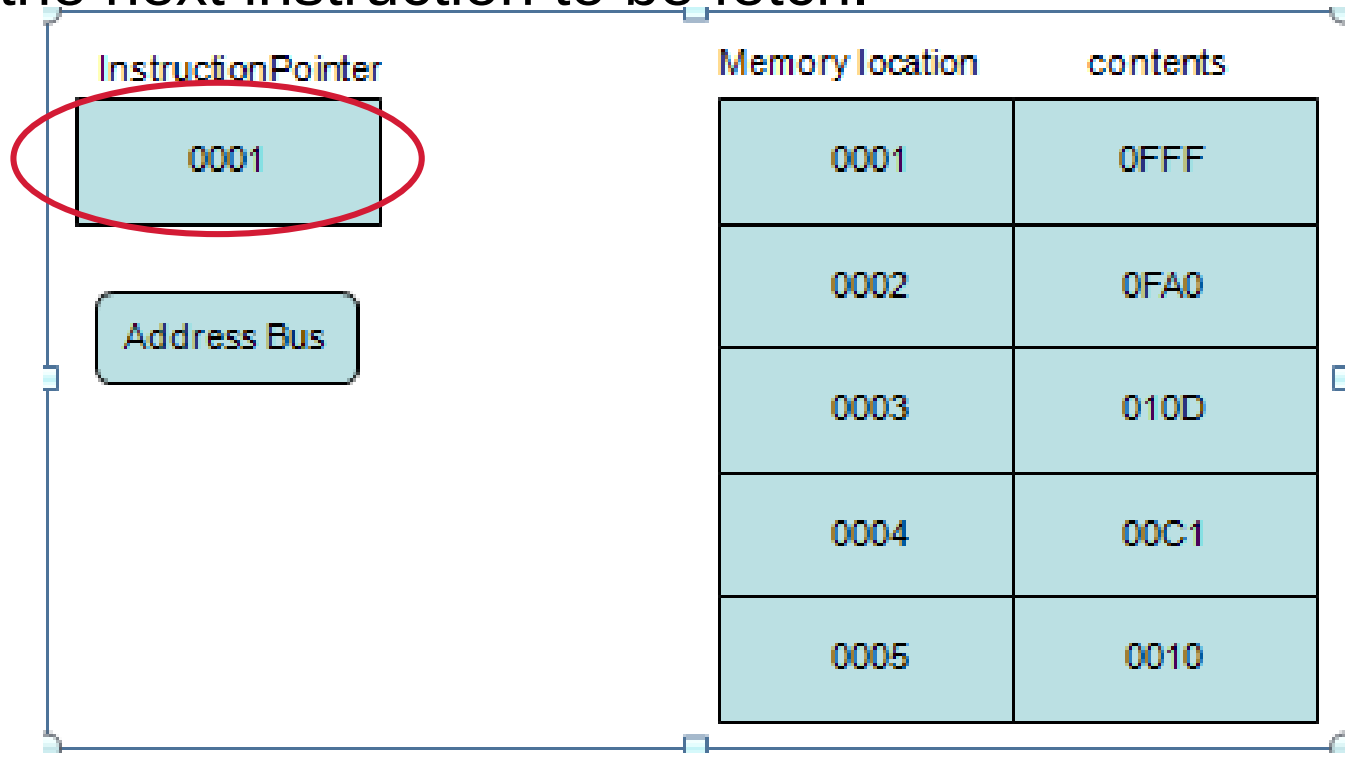
3. **Memory data (access) register** **(MDR):** A two-way register that holds data fetched from memory (and ready for the CPU to process) or data waiting to be stored in memory. (This is also known as the memory buffer register (MBR).)

- **4. Instruction register (IR):** A temporary holding ground for the instruction that has just been fetched from memory.
- 5. **Control unit (CU):** Decodes the program instruction in the IR, selecting machine resources, such as a data source register and a particular arithmetic operation, and coordinates activation of those resources.
- 6. **Arithmetic logic unit (ALU):** Performs mathematical and logical operations.

Fetching an instruction

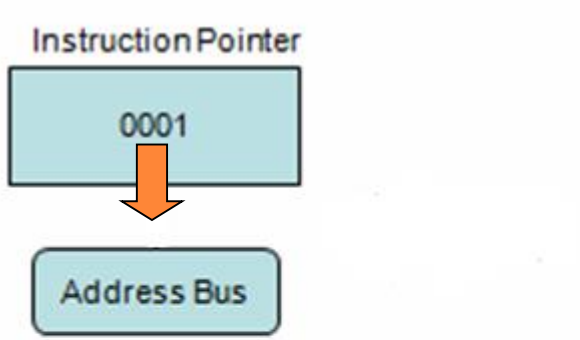
- **Step 1**

 Instruction pointer (program counter) hold the address of the next instruction to be fetch.



FETCHING AN INSTRUCTION (cont.)

- **Step 2**

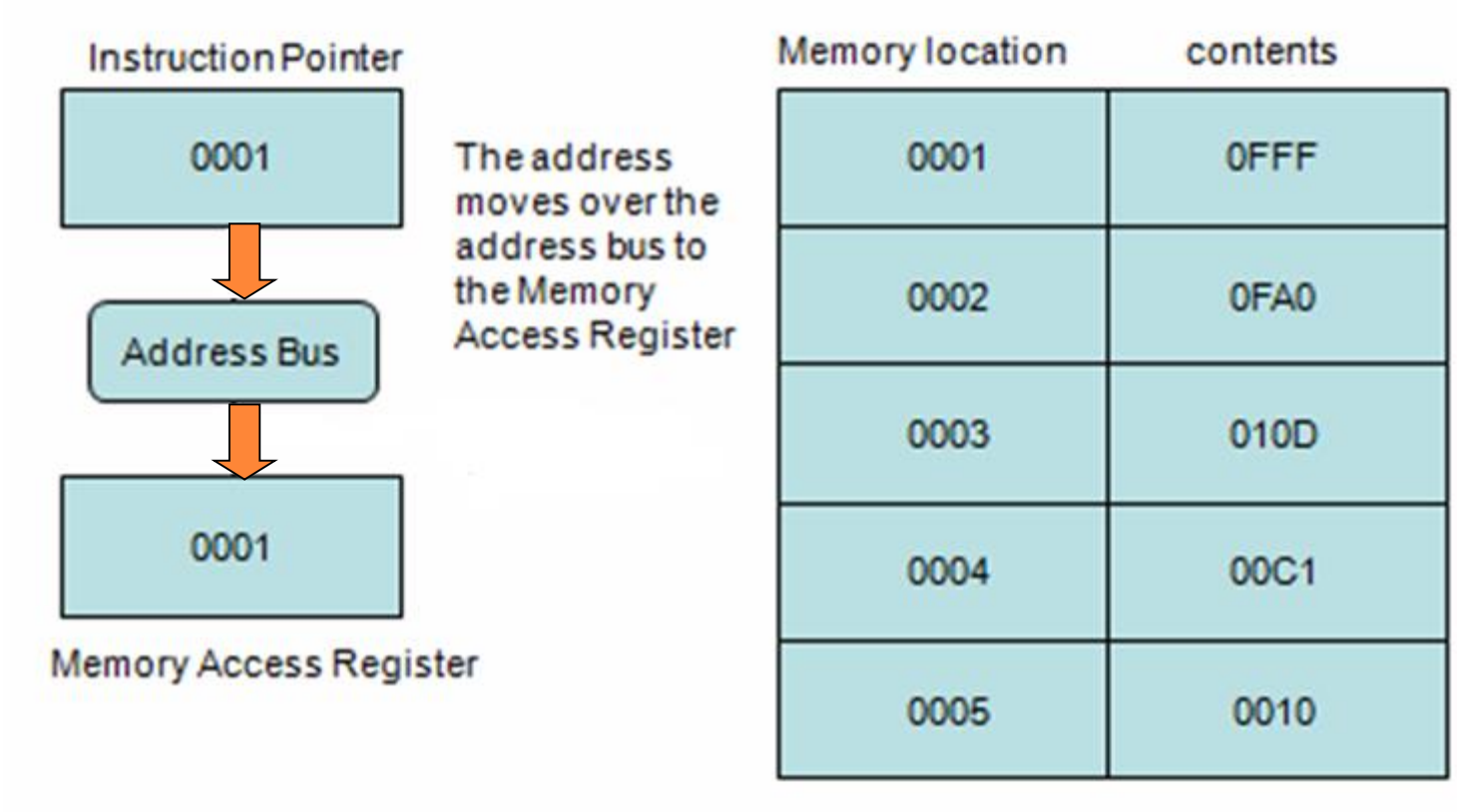


Contents of the Program Counter are passed across the Address Bus

Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

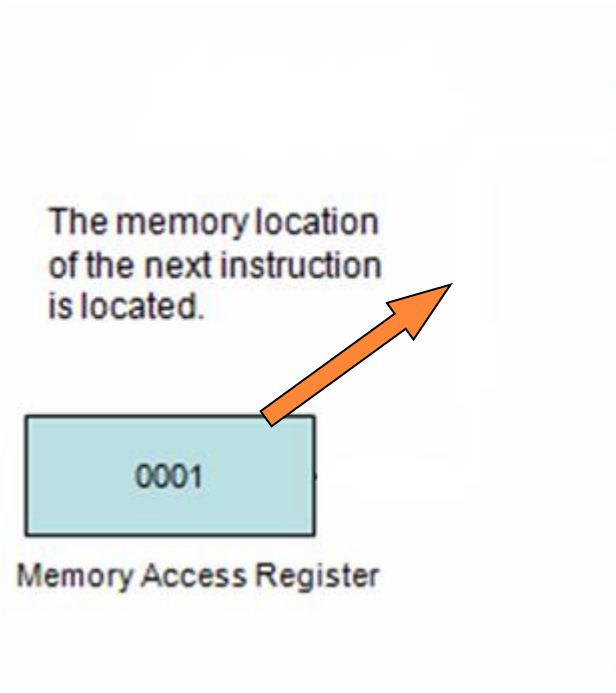
FETCHING AN INSTRUCTION (cont.)

- **Step 3**



FETCHING AN INSTRUCTION (cont.)

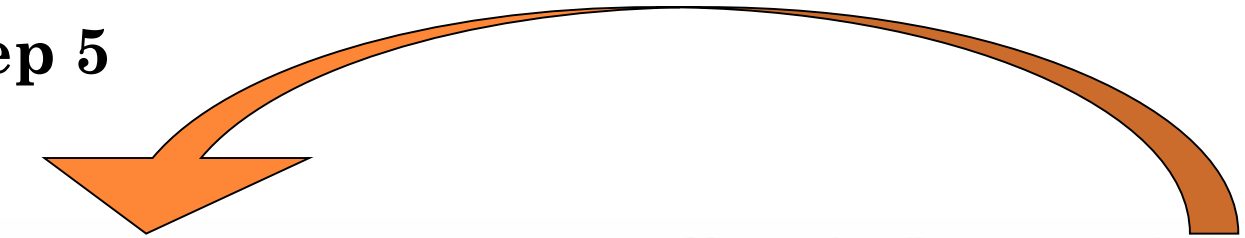
- **Step 4**



Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

FETCHING AN INSTRUCTION (cont.)

- **Step 5**



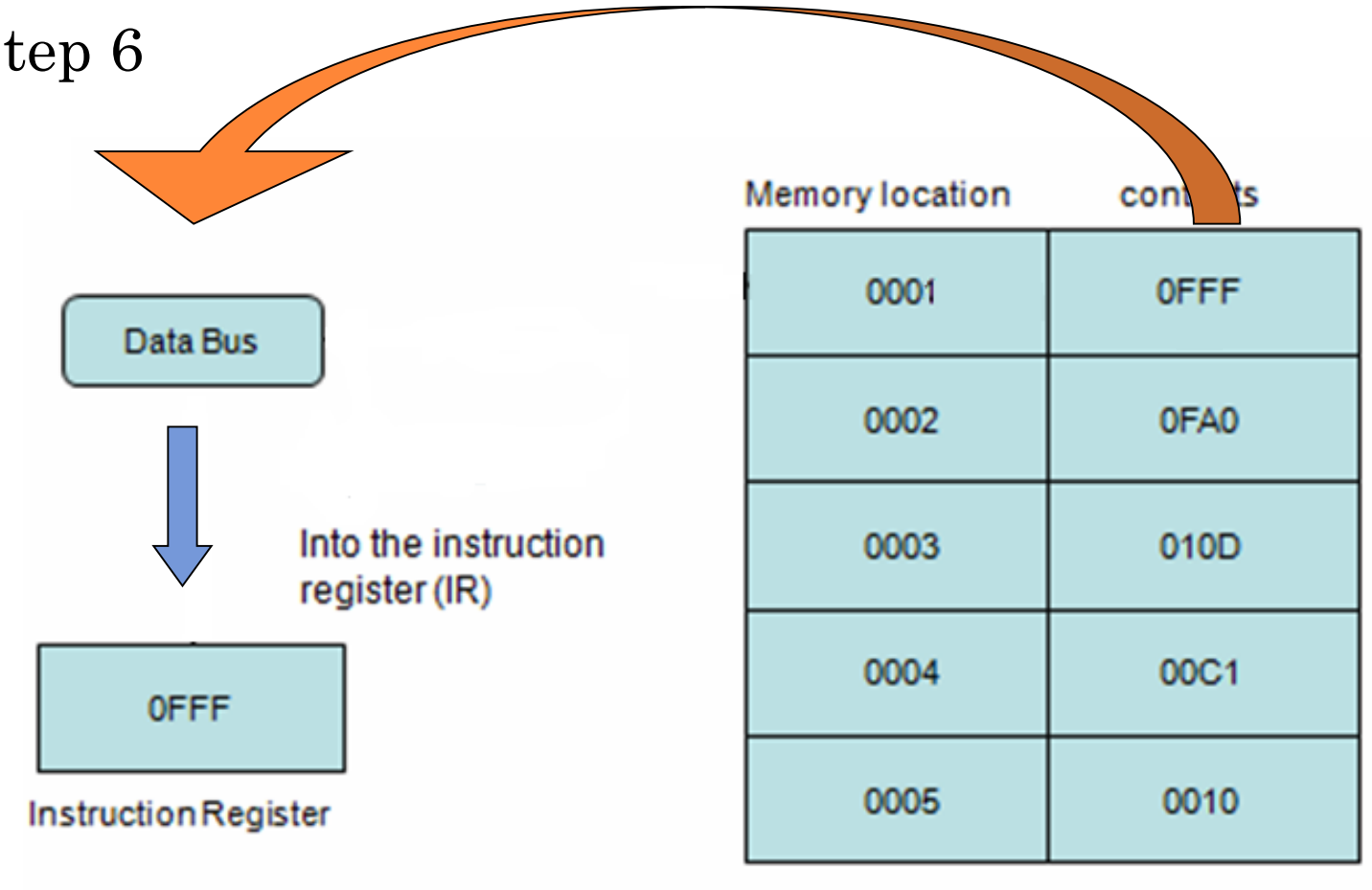
Data Bus

The contents of memory at the given location are moved across the data bus

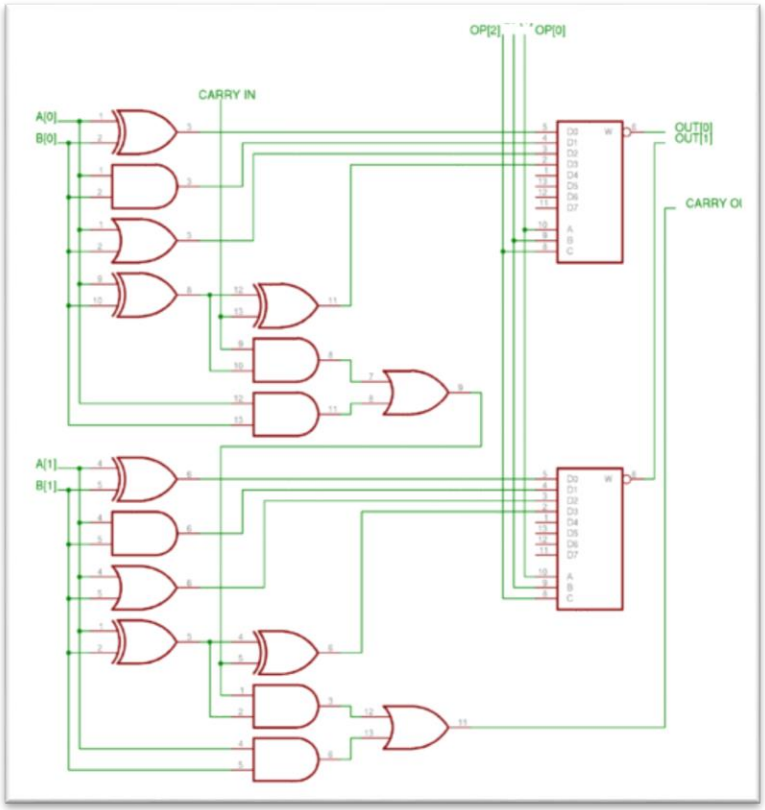
Memory location	contents
0001	0FFF
0002	0FA0
0003	010D
0004	00C1
0005	0010

FETCHING AN INSTRUCTION (cont.)

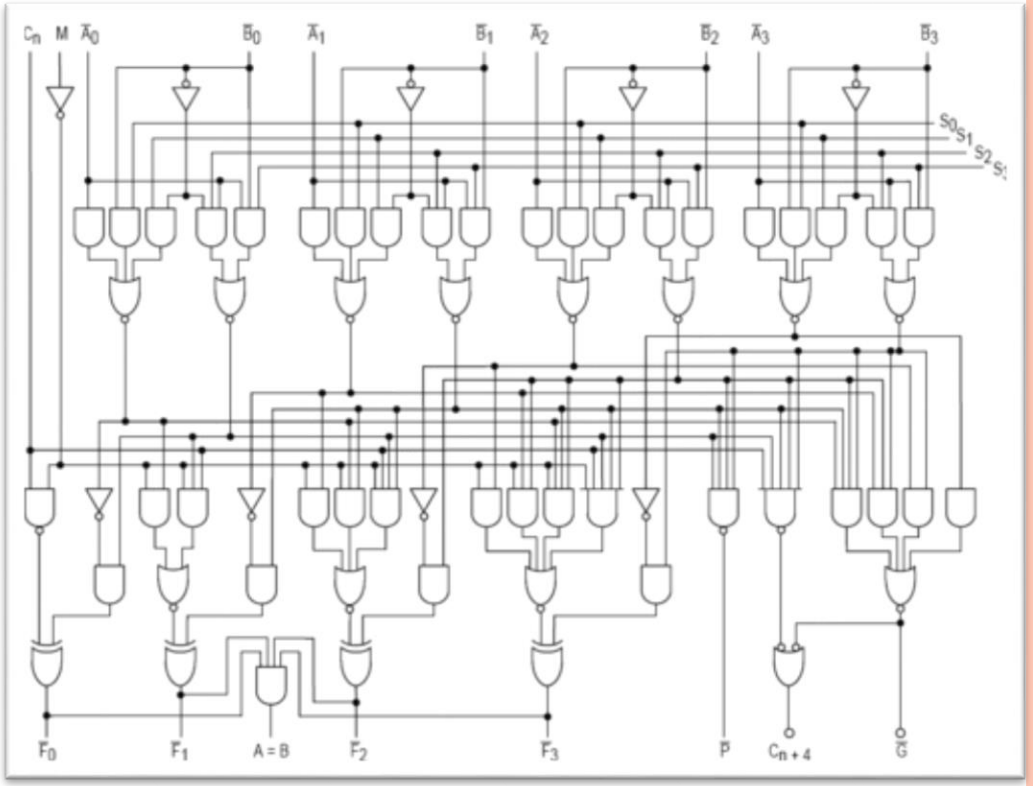
- Step 6



Internal structure of ALU

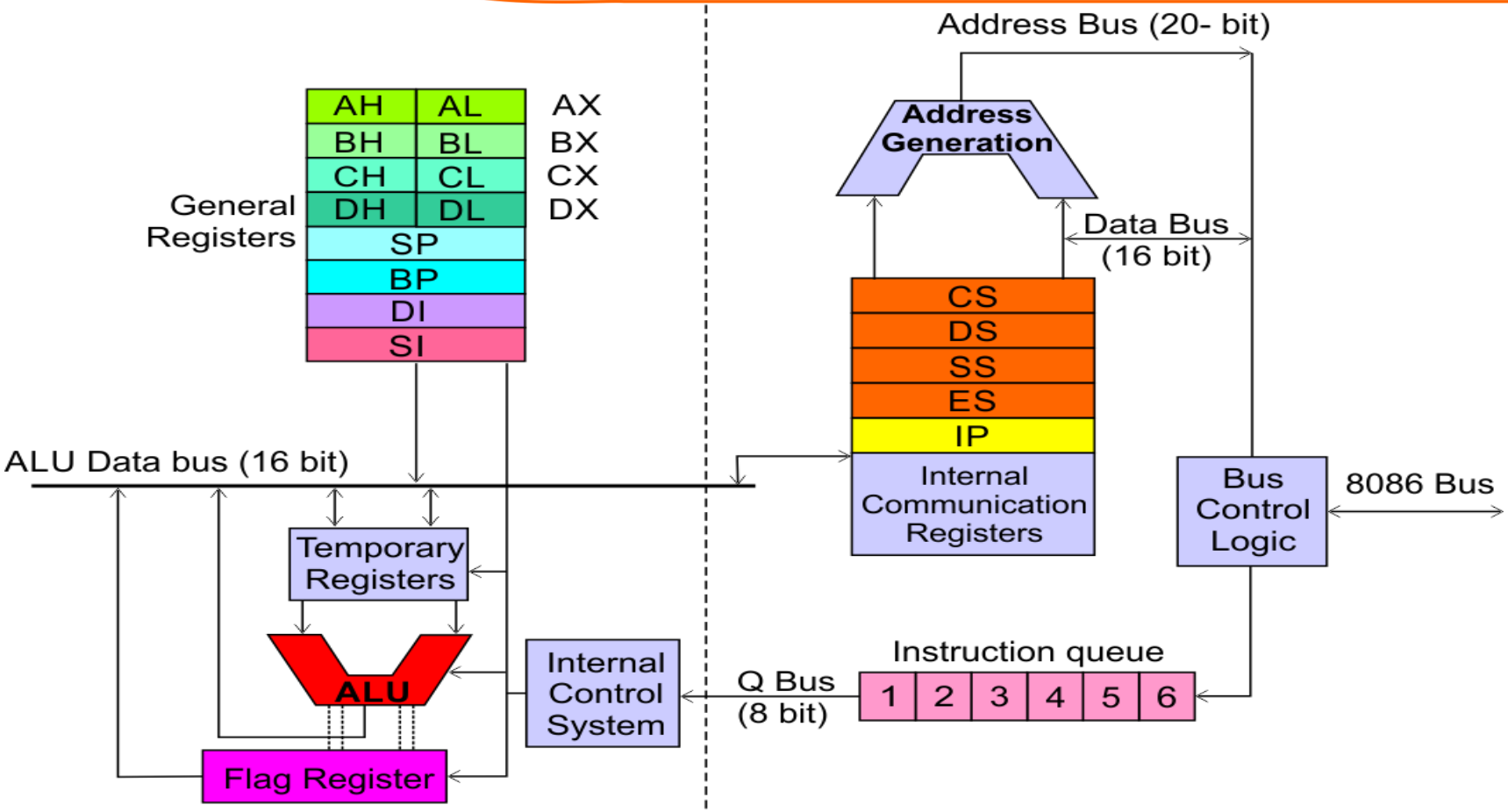


2 bits of ALU



4 bits of ALU

Architecture with more details



Execution Unit (EU)

EU executes instructions that have already been fetched by the BIU.

BIU and EU functions separately.

Bus Interface Unit (BIU)

BIU fetches instructions, reads data from memory and I/O ports, writes data to memory and I/ O ports.

Internal architecture of 8086

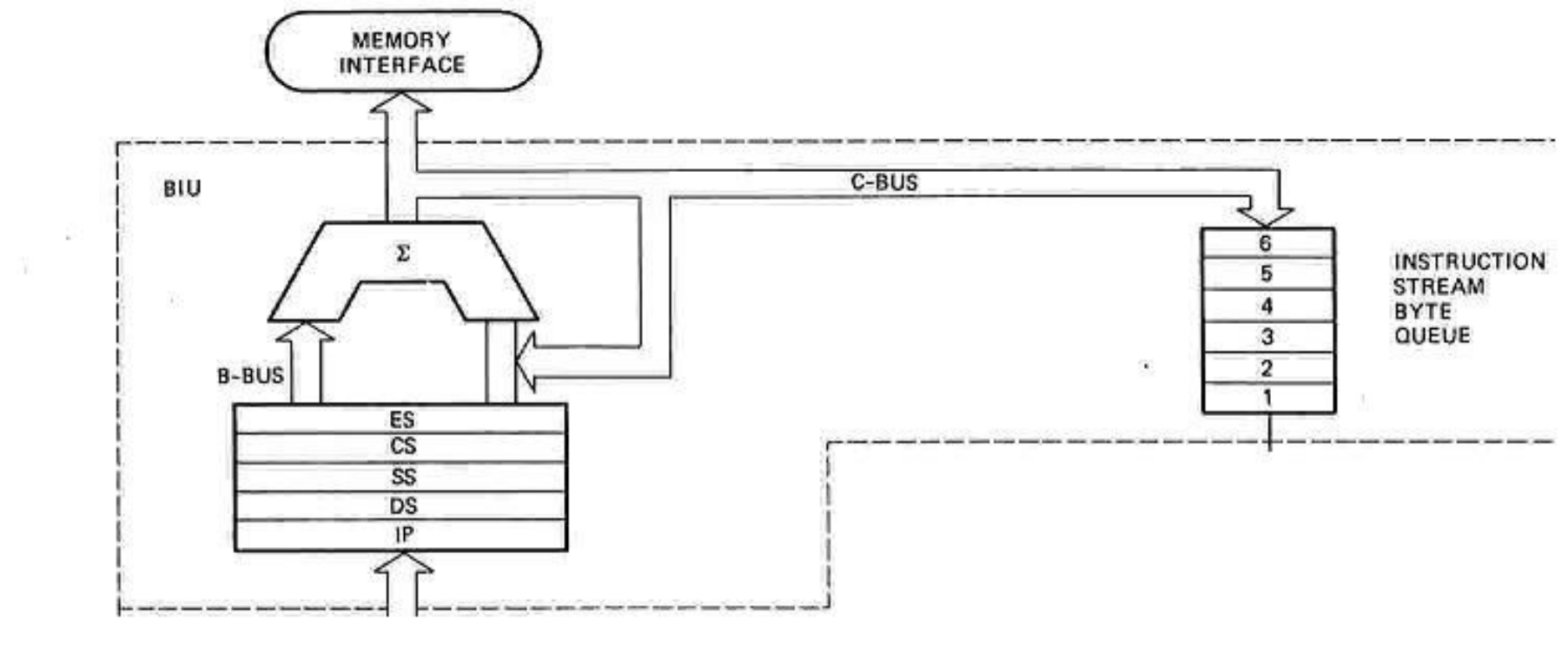
- 8086 has two blocks BIU and EU.
- The BIU handles all transactions of data and addresses on the buses for EU.
- The BIU performs all bus operations such as instruction fetching, reading and writing operands for memory and calculating the addresses of the memory operands. The instruction bytes are transferred to the instruction queue.
- EU executes instructions from the instruction system byte queue.

- BIU contains Instruction queue, Segment registers, Instruction pointer, Address adder.
- EU contains Control circuitry, Instruction decoder, ALU, Pointer and Index register, Flag register.

BUS INTERFACE UNIT (BIU)

Contains

- **6-byte Instruction Queue (Q)**
- **The Segment Registers (CS, DS, ES, SS).**
- **The Instruction Pointer (IP).**
- **The Address Summing block (Σ)**

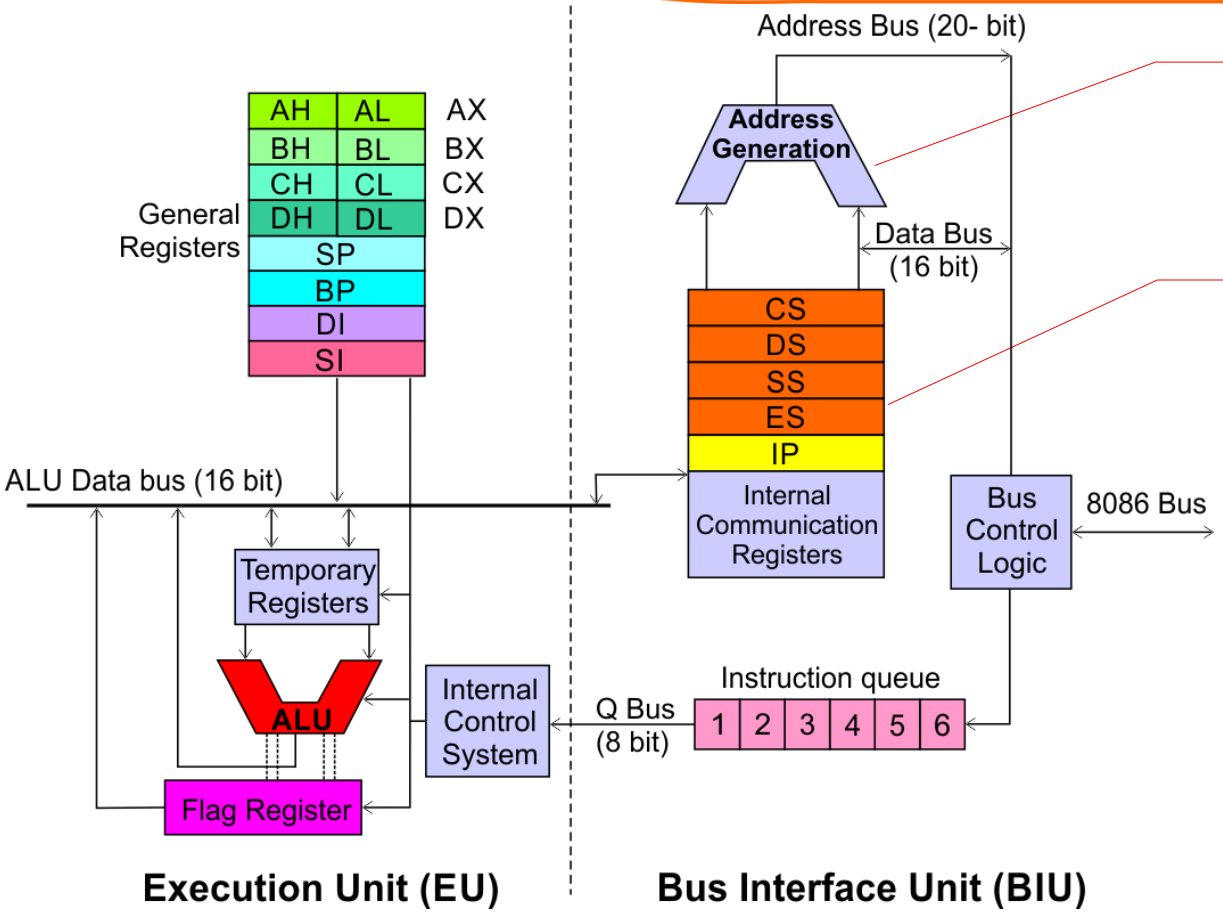


THE QUEUE (Q)

- The BIU uses a mechanism known as an **instruction stream queue** to implement a ***pipeline architecture.***
- This queue permits pre-fetch of up to **6 bytes** of instruction code. Whenever the queue of the BIU is not full, it has room for at least two more bytes and at the same time the EU is not requesting it to read or write operands from memory, the BIU is free to look ahead in the program by pre-fetching the next sequential instruction.

Architecture

Bus Interface Unit (BIU)



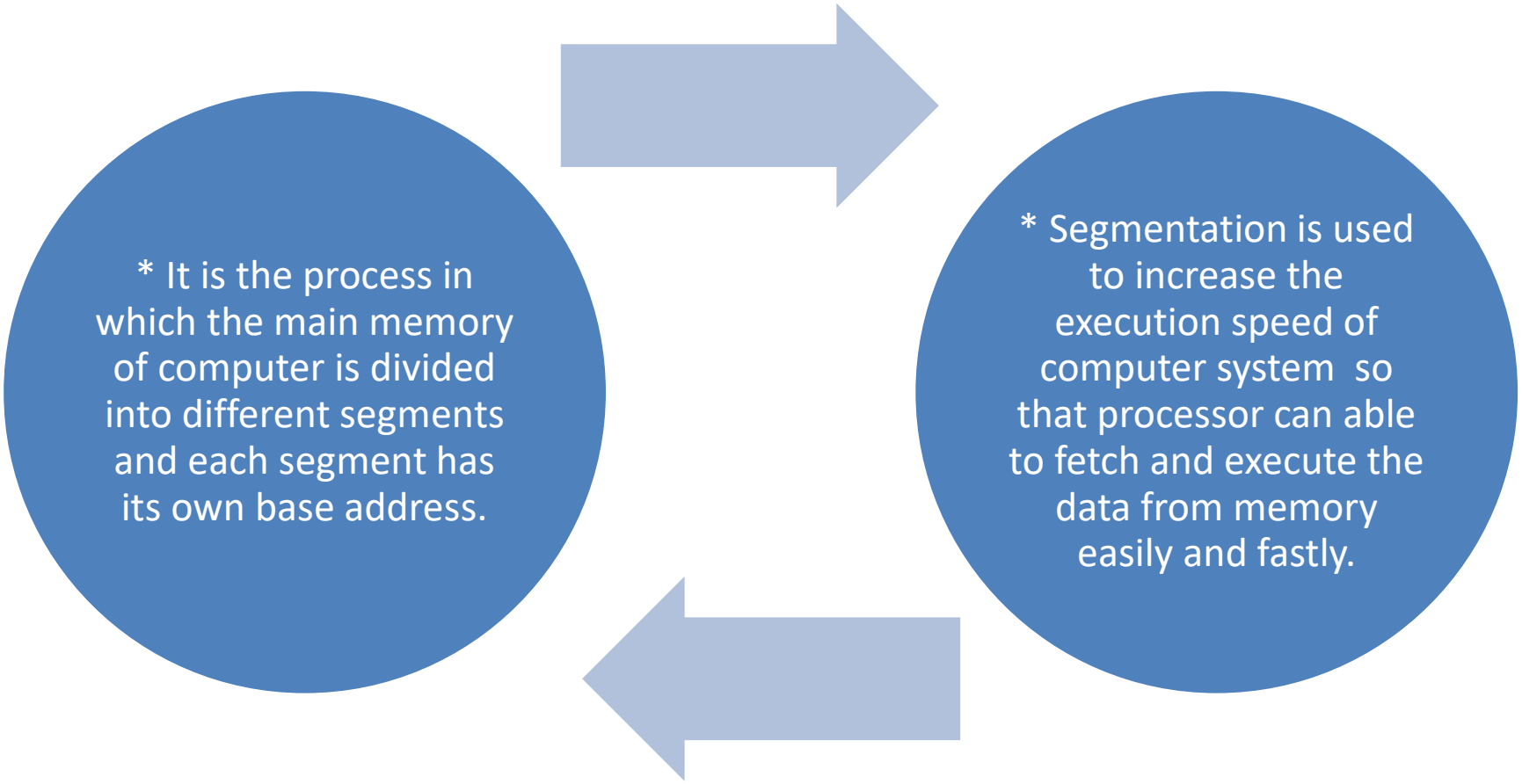
Dedicated Adder to generate 20 bit address

Four 16-bit segment registers

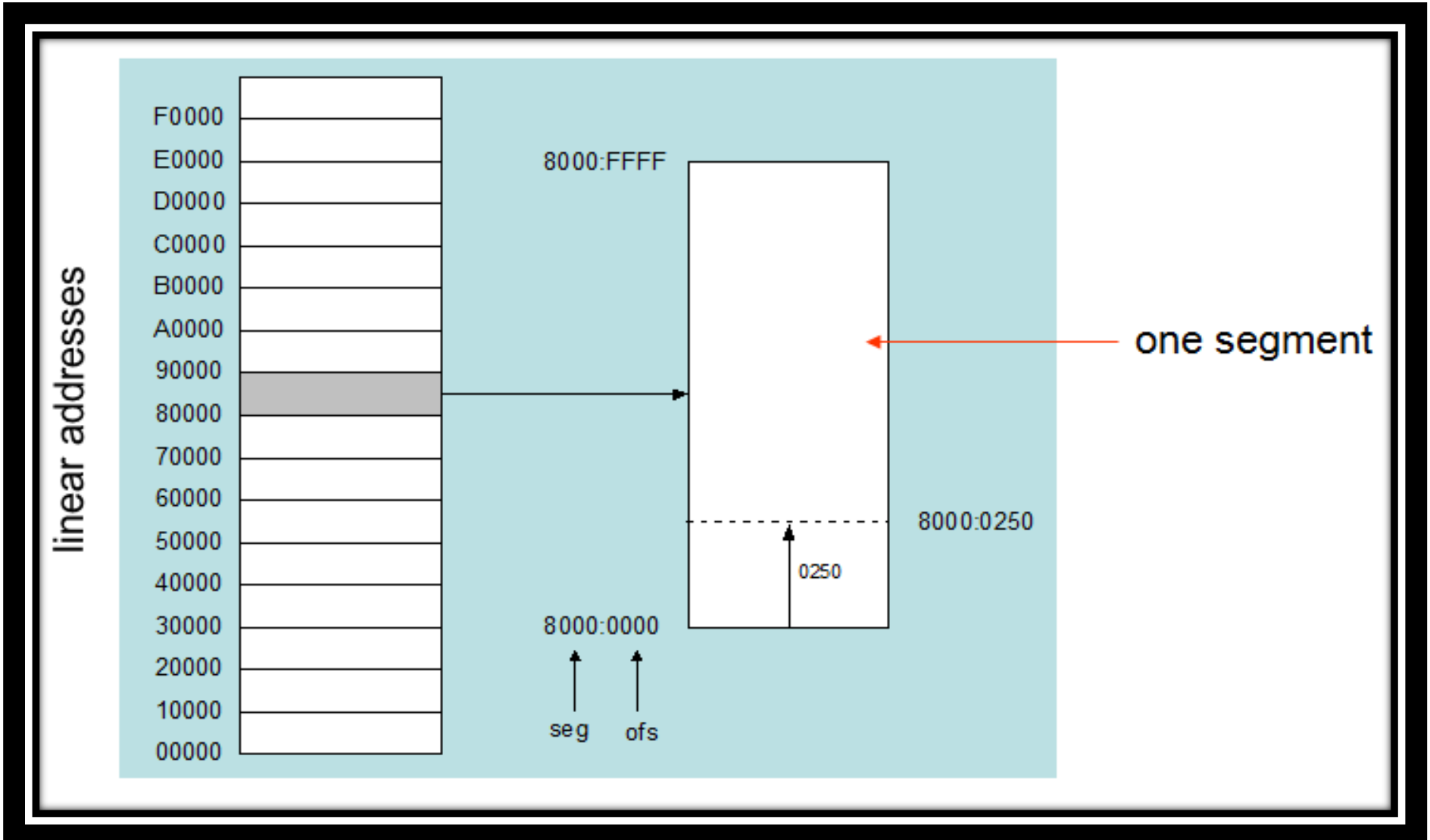
Code Segment (CS)
Data Segment (DS)
Stack Segment (SS)
Extra Segment (ES)

Memory Segmentation

Segmentation



Segmentation



Segmentation in 8086

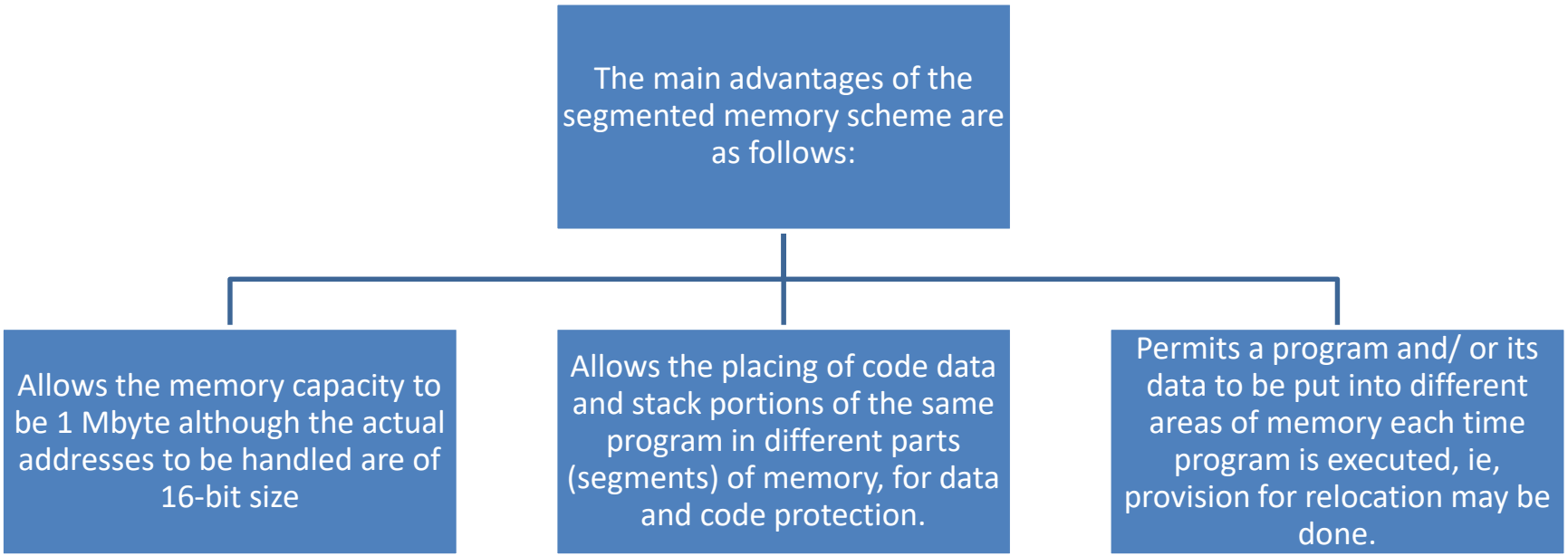
The size of address bus of 8086 is 20 and is able to address 1 Mbytes () of physical memory.

The compete 1 Mbytes memory can be divided into 16 segments, each of 64 Kbytes size.

The addresses of the segment may be assigned as 0000*H* to *F*000*H* respectively.

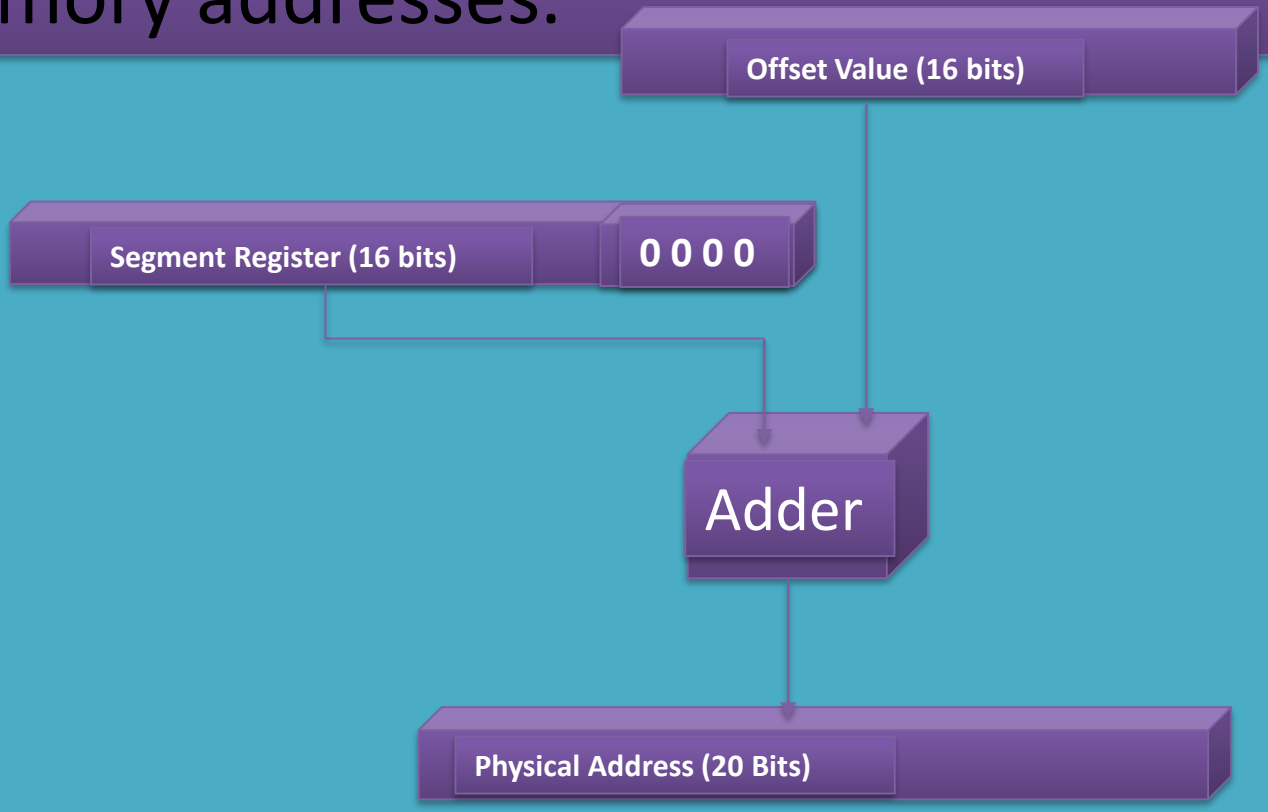
The offset values are from 0000*H* to *FFFFFH*.

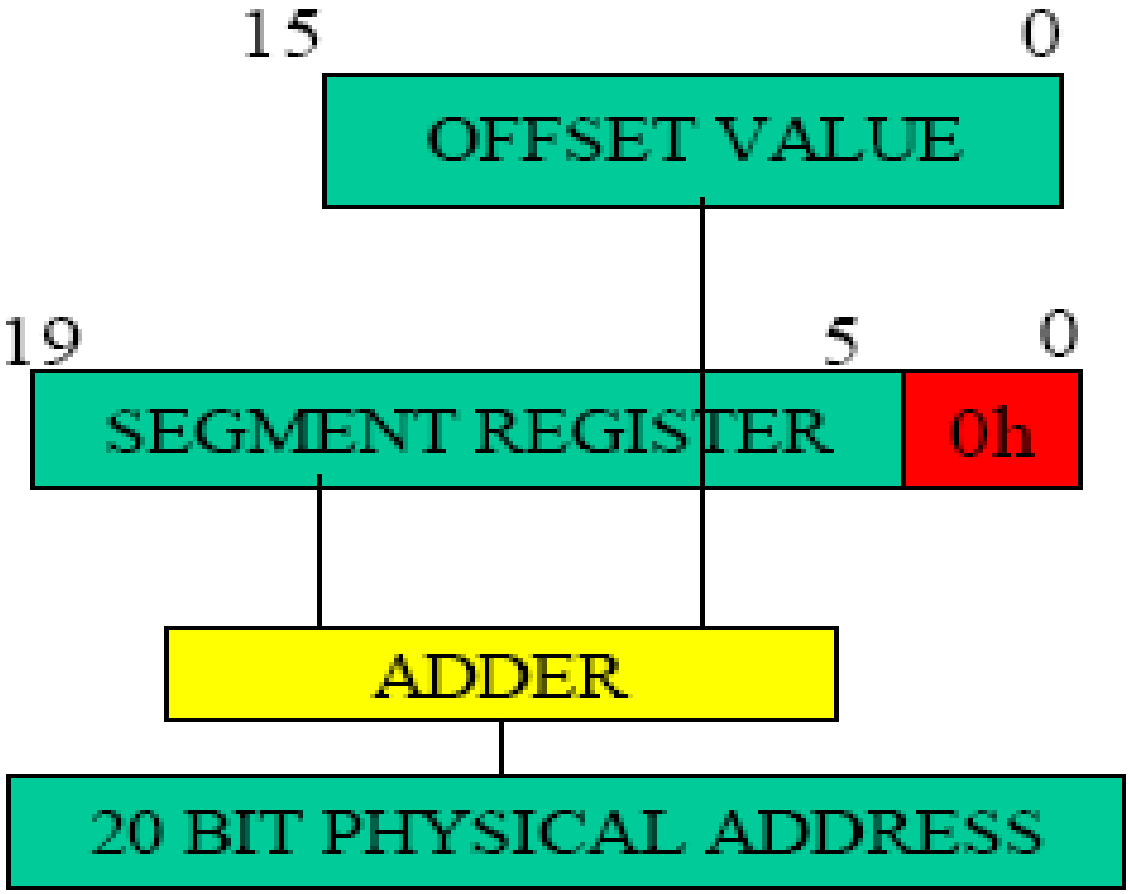
Advantages of segmentation



Memory Address Generation

- The BIU has a dedicated adder for determining physical memory addresses.



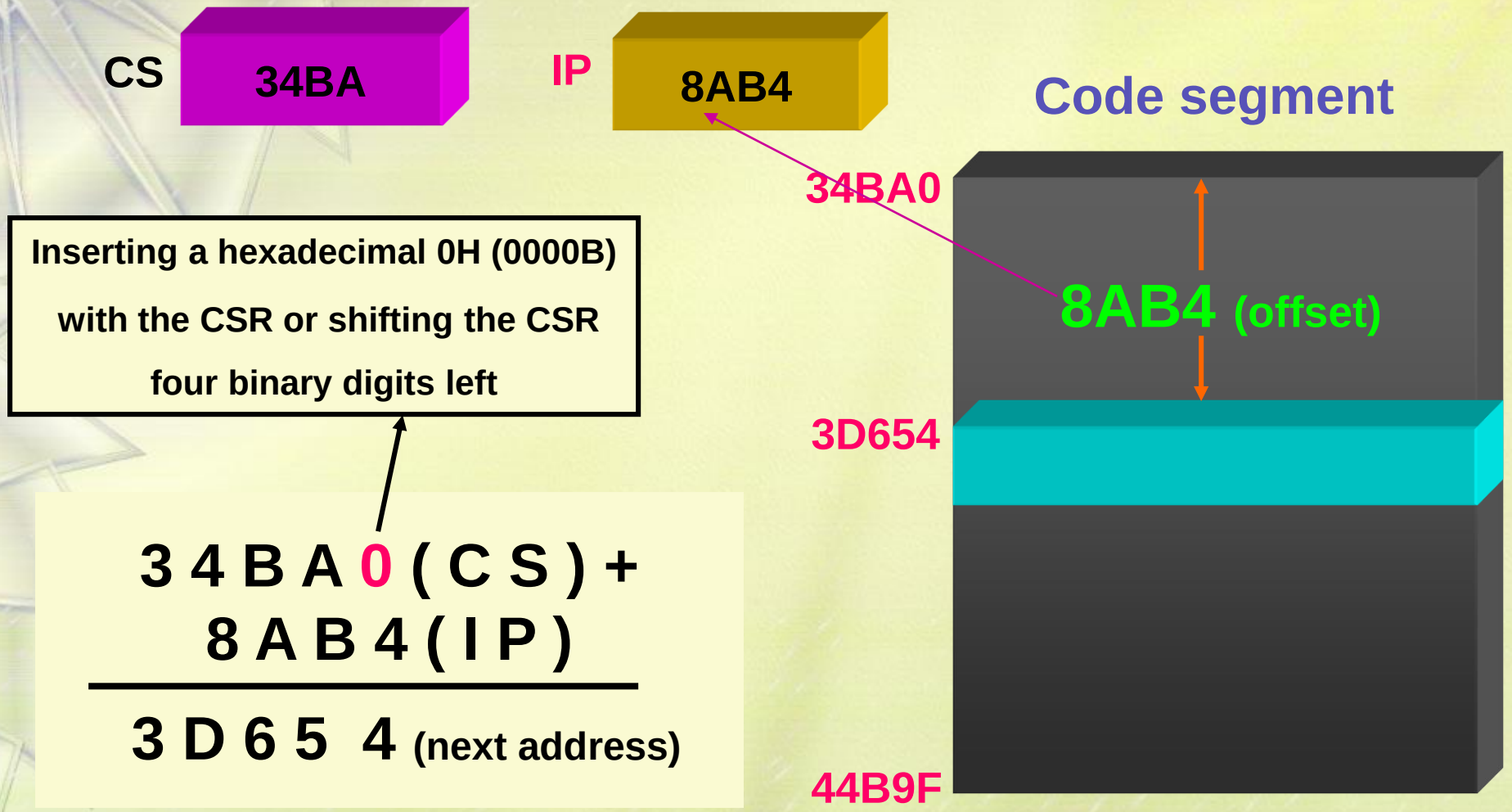

15
0
OFFSET VALUE
19
5
0
SEGMENT REGISTER
0h
ADDER
20 BIT PHYSICAL ADDRESS

Segment : Offset Address

- Logical Address is specified as **segment: offset**
- Physical address is obtained by shifting the segment address 4 bits to the left and adding the offset address.
- Thus the physical address of the logical address **A4FB:4872** is:

```
  A4FB0
+ 4872
  A9822
```

- **The following examples shows the CS:IP scheme of address formation:**



Segments, Segment Registers & Offset Registers

4 Segments in 8086

- Code Segment (CS)
- Data Segment (DS)
- Stack Segment (SS)
- Extra Segment (ES)

SEGMENT	SEGMENT REGISTER	OFFSET REGISTER
Code Segment	CSR	**Instruction Pointer (IP)**
Data Segment	DSR	**Source Index (SI)**
Extra Segment	ESR	**Destination Index (DI)**
Stack Segment	SSR	**Stack Pointer (SP) / Base Pointer (BP)**

Segment and Address register combination

- CS:IP
- SS:SP SS:BP
- DS:BX DS:SI
- DS:DI (for other than string operations)
- ES:DI (for string operations)

Segments, Segment Registers & Offset Registers

Segment Size = 64KB

Maximum number of segments possible = 14

Logical Address – 16 bits

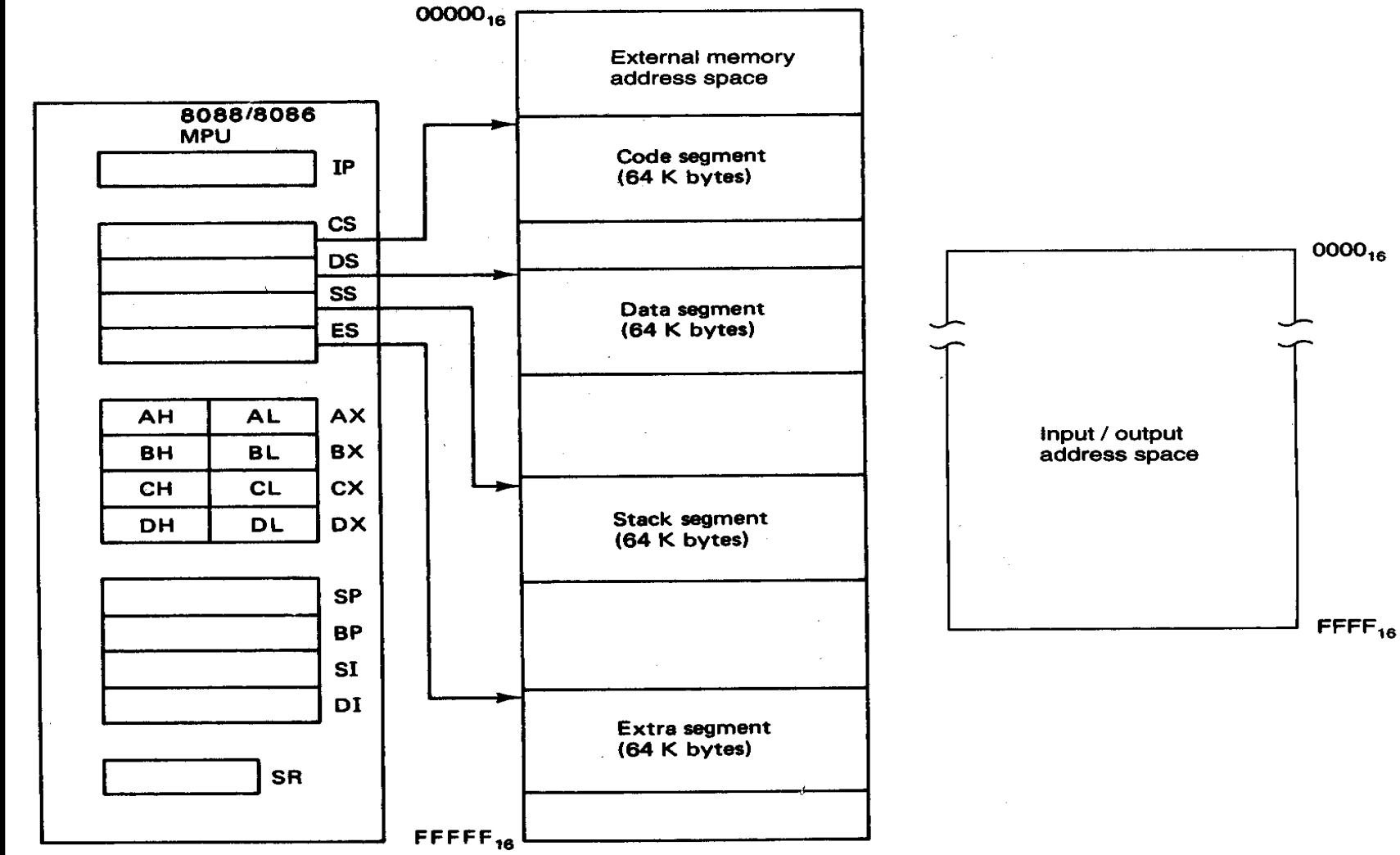
Physical Address – 20 bits

2 Logical Addresses for each Segments.

- Base Address (16 bits)
- Offset Address (16 bits)

Segment registers are used to store the Base address of the segment.

Segmented Memory Representation



- The size of each segment is 64 KB
- A segment may be located any where in the memory
- Each of these segments can be used for a specific function.
 - Code segment is used for storing the instructions.
 - The stack segment is used as a stack and it is used to store the return addresses.
 - The data and extra segments are used for storing data byte.

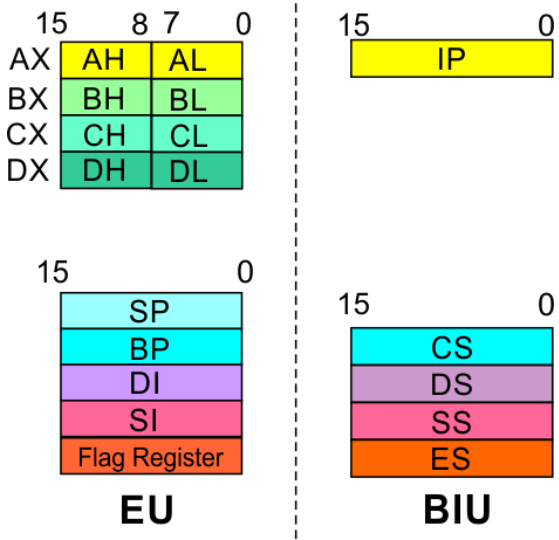
Segment registers

- In 8086/88 the processors have 4 segments registers
- Code Segment register (CS), Data Segment register (DS), Extra Segment register (ES) and Stack Segment (SS) register.
- All are 16 bit registers.
- Each of the Segment registers store the upper 16 bit address of the starting address of the corresponding segments.

Segment Registers

Code Segment Register

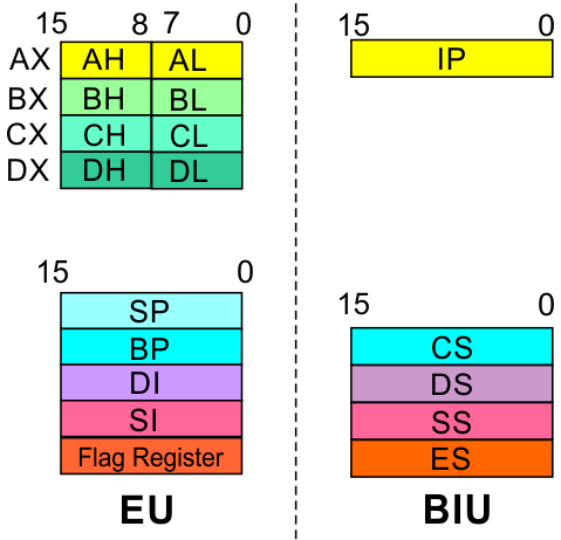
- 16-bit
- CS contains the base or start of the current code segment; IP contains the distance or offset from this address to the next instruction byte to be fetched.
- BIU computes the 20-bit physical address by logically shifting the contents of CS 4-bits to the left and then adding the 16-bit contents of IP.
- That is, all instructions of a program are relative to the contents of the CS register multiplied by 10 and then offset is added provided by the IP.



Segment Registers

Data Segment Register

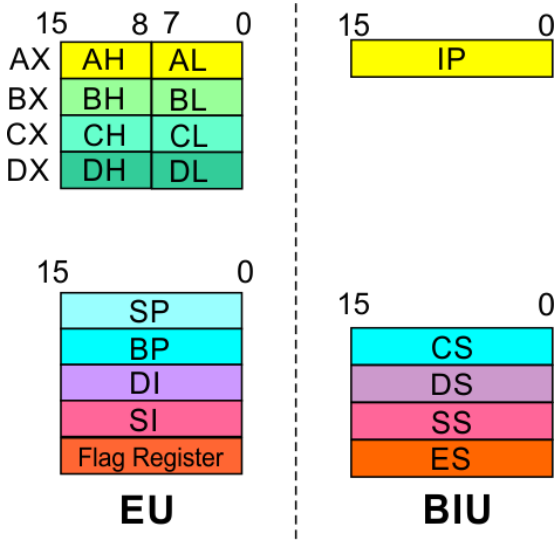
- 16-bit
- Points to the current data segment; operands for most instructions are fetched from this segment.
- The 16-bit contents of the Source Index (SI) or Destination Index (DI) or a 16-bit displacement are used as offset for computing the 20-bit physical address.



Segment Registers

Stack Segment Register

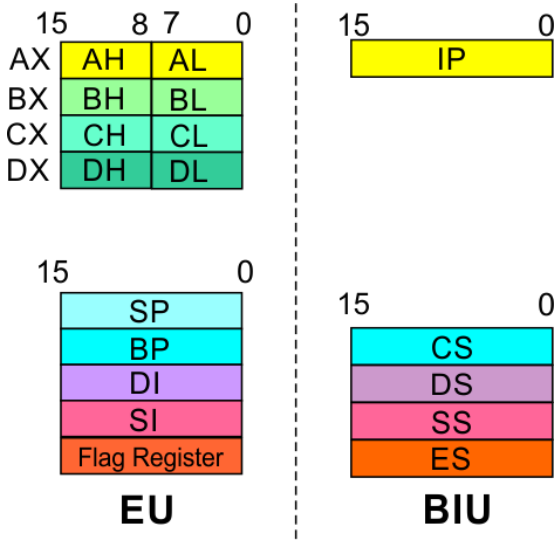
- **16-bit**
- **Points to the current stack.**
- **The 20-bit physical stack address is calculated from the Stack Segment (SS) and the Stack Pointer (SP) for stack instructions such as PUSH and POP.**
- **In based addressing mode, the 20-bit physical stack address is calculated from the Stack segment (SS) and the Base Pointer (BP).**



Segment Registers

Extra Segment Register

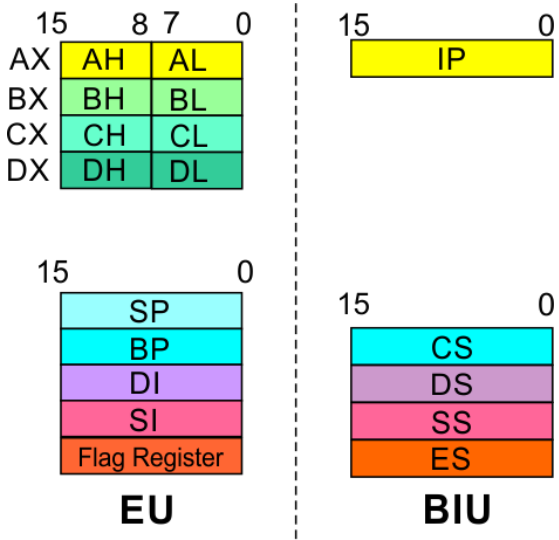
- 16-bit
- Points to the extra segment in which data (in excess of 64K pointed to by the DS) is stored.
- String instructions use the ES and DI to determine the 20-bit physical address for the destination.



Segment Registers

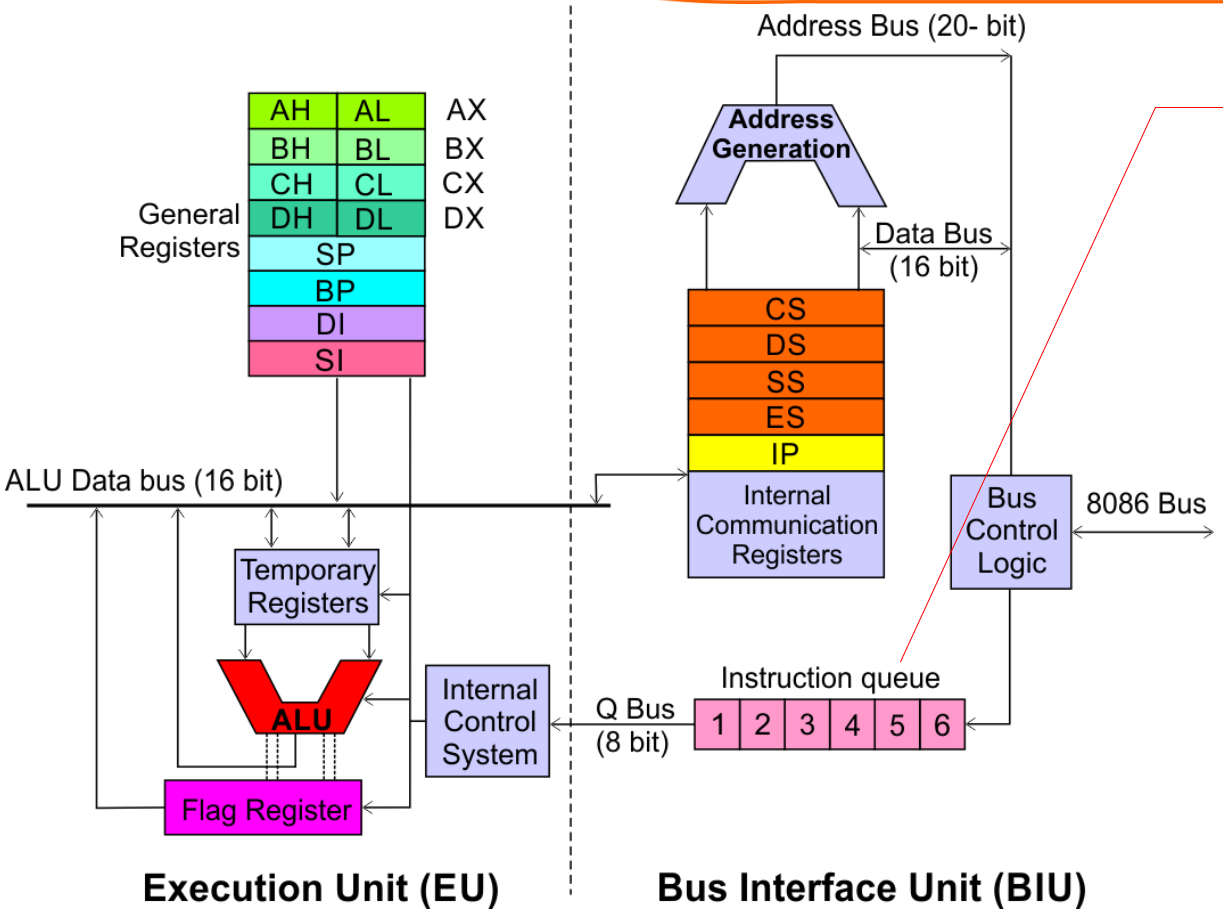
Instruction Pointer

- 16-bit
- Always points to the next instruction to be executed within the currently executing code segment.
- So, this register contains the 16-bit offset address pointing to the next instruction code within the 64Kb of the code segment area.
- Its content is automatically incremented as the execution of the next instruction takes place.



Architecture

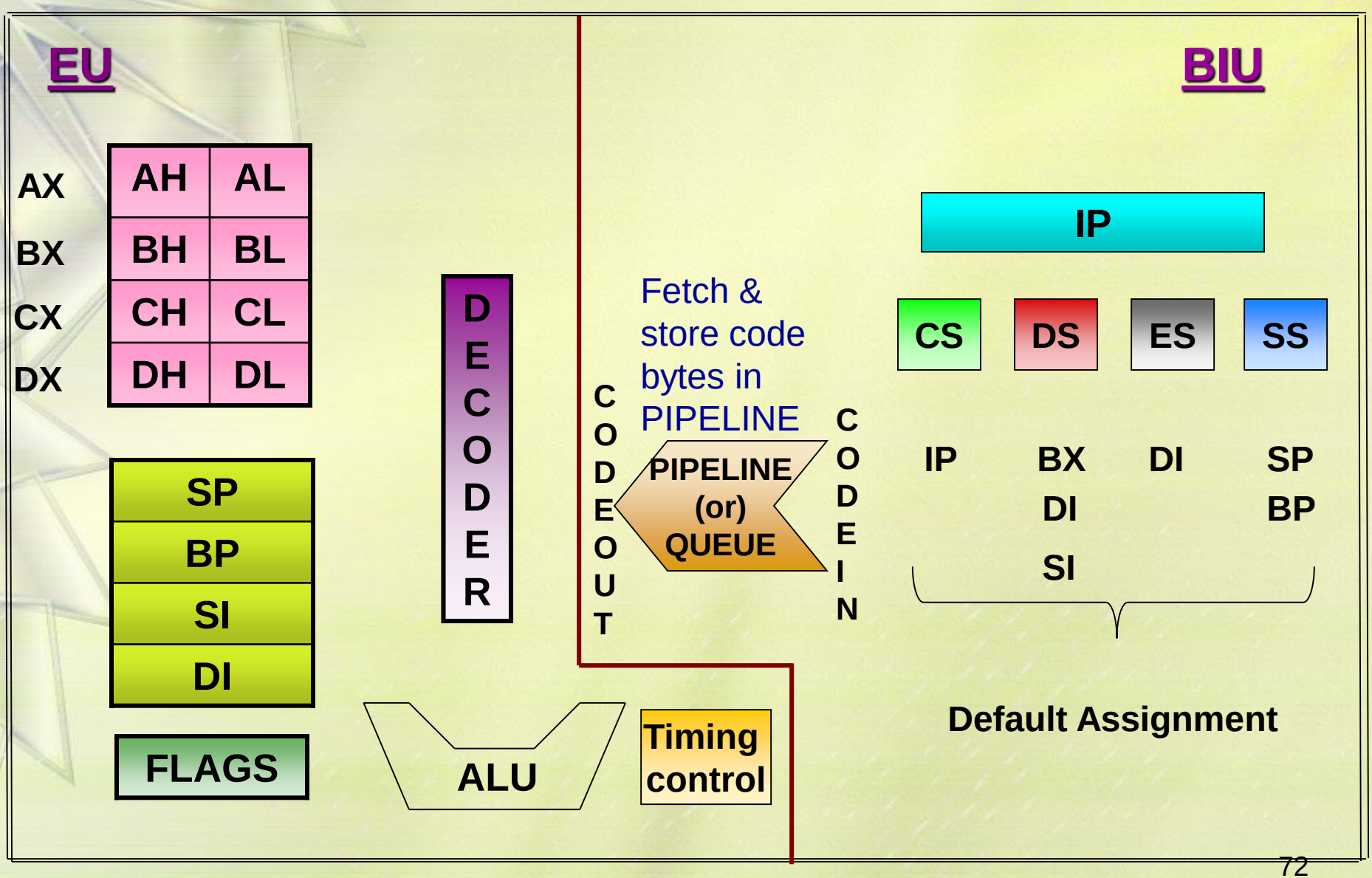
Bus Interface Unit (BIU)



Instruction queue

- **A group of First-In-First-Out (FIFO) in which up to 6 bytes of instruction code are pre fetched from the memory ahead of time.**
- **This is done in order to speed up the execution by overlapping instruction fetch with execution.**
- **This mechanism is known as pipelining.**

Summary of Registers & Pipeline of 8086 µP



EXECUTION UNIT

- **Decodes instructions fetched by the BIU**
- **Generate control signals,**
- **Executes instructions.**

The main parts are:

- **Control Circuitry**
- **Instruction decoder**
- **ALU**

Architecture

Execution Unit (EU)

EU decodes and executes instructions.

A decoder in the EU control system translates instructions.

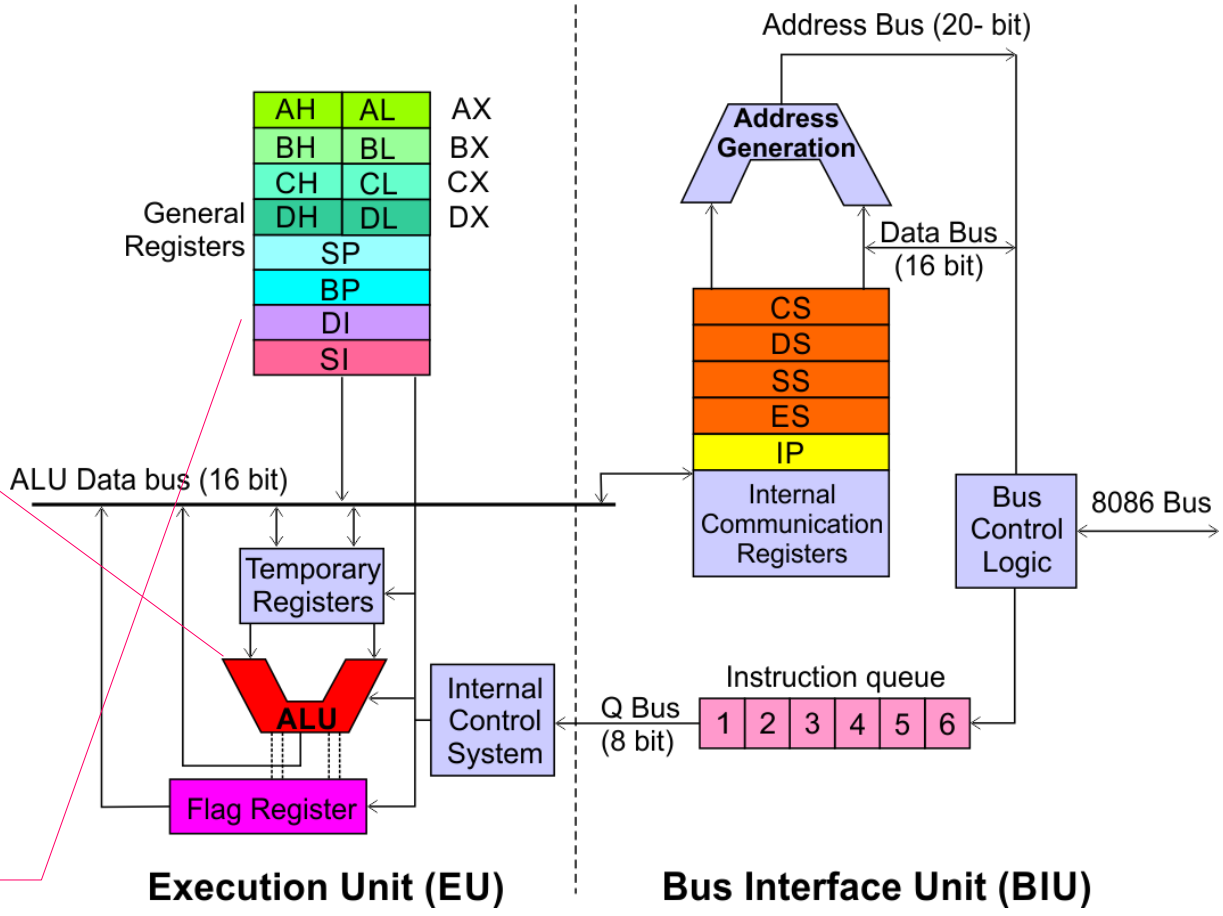
16-bit ALU for performing arithmetic and logic operation

Four general purpose registers(AX, BX, CX, DX);

Pointer registers (Stack Pointer, Base Pointer);

and

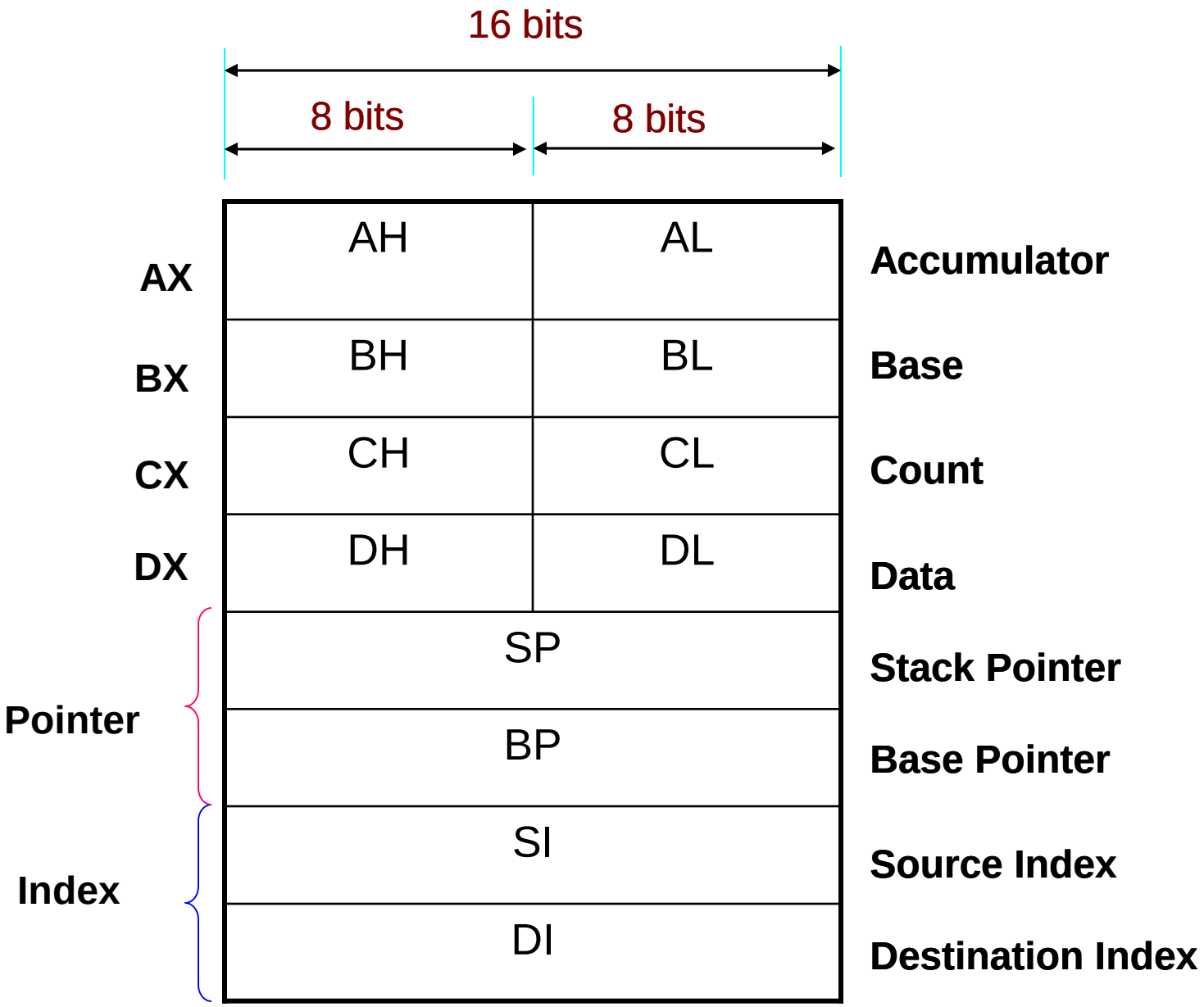
Index registers (Source Index, Destination Index) each of 16-bits



Some of the 16 bit registers can be used as two 8 bit registers as :

AX can be used as AH and AL
BX can be used as BH and BL
CX can be used as CH and CL
DX can be used as DH and DL

EXECUTION UNIT – General Purpose Registers



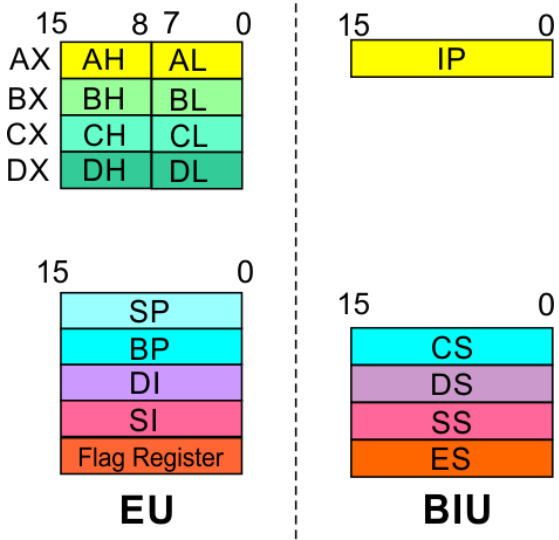
ACCUMULATOR

- a register in which intermediate arithmetic and logic results are stored.
- example for accumulator use is summing a list of numbers.
 - The accumulator is initially set to zero, then each number in turn is added to the value in the accumulator.
 - Only when all numbers have been added is the result held in the accumulator written to main memory or to another, non-accumulator, CPU register.

EU Registers

Accumulator Register (AX)

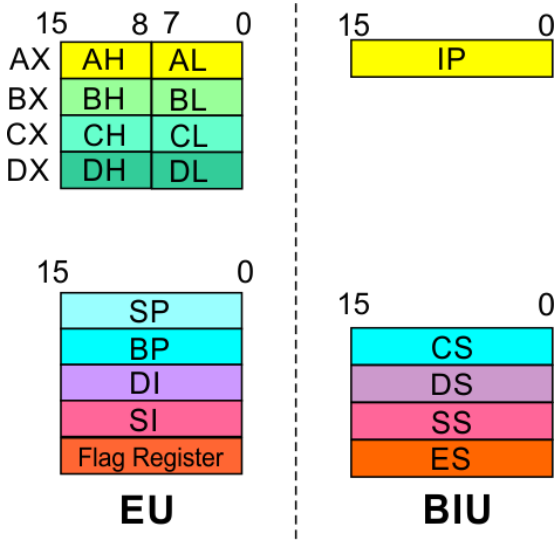
- Consists of two 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX.
- AL in this case contains the low order byte of the word, and AH contains the high-order byte.
- The I/O instructions use the AX or AL for inputting / outputting 16 or 8 bit data to or from an I/O port.
- Multiplication and Division instructions also use the AX or AL.



EU Registers

Base Register (BX)

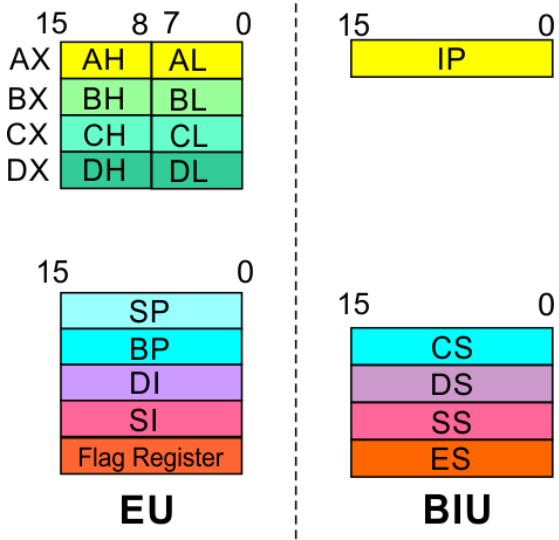
- Consists of two 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX.
- BL in this case contains the low-order byte of the word, and BH contains the high-order byte.
- This is the only general purpose register whose contents can be used for addressing the 8086 memory.
- All memory references utilizing this register content for addressing use DS as the default segment register.



EU Registers

Counter Register (CX)

- Consists of two 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX.
- When combined, CL register contains the low order byte of the word, and CH contains the high-order byte.
- Instructions such as SHIFT, ROTATE and LOOP use the contents of CX as a counter.98



Example:

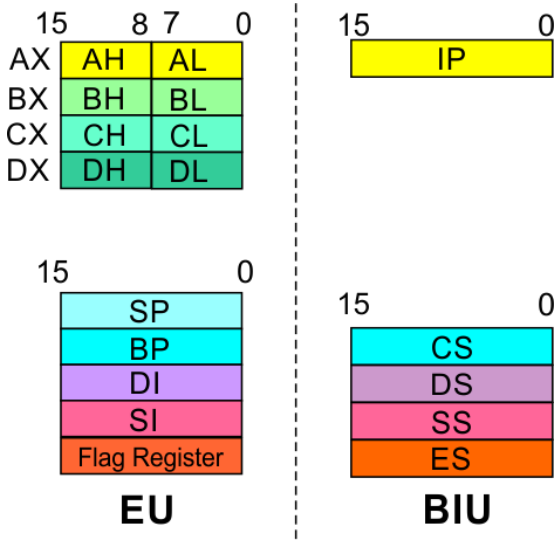
The instruction LOOP START automatically decrements CX by 1 without affecting flags and will check if [CX] = 0.

If it is zero, 8086 executes the next instruction; otherwise the 8086 branches to the label START.

EU Registers

Data Register (DX)

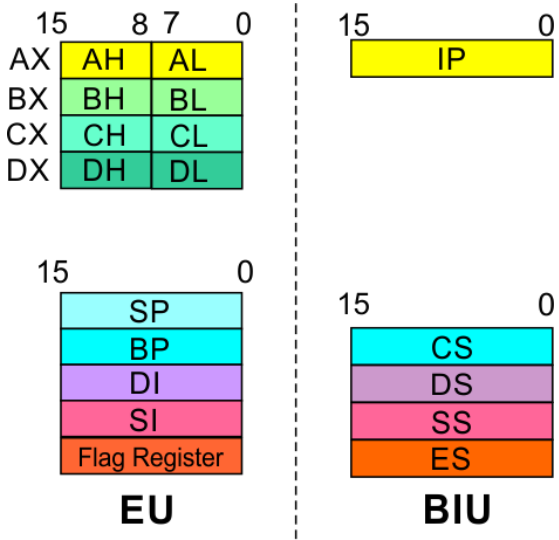
- Consists of two 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX.
- When combined, DL register contains the low order byte of the word, and DH contains the high-order byte.
- Used to hold the high 16-bit result (data) in 16 X 16 multiplication or the high 16-bit dividend (data) before a 32 ÷ 16 division and the 16-bit reminder after division.



EU Registers

Stack Pointer (SP) and Base Pointer (BP)

- SP and BP are used to access data in the stack segment.
- SP is used as an offset from the current SS during execution of instructions that involve the stack segment in the external memory.
- SP contents are automatically updated (incremented/ decremented) due to execution of a POP or PUSH instruction.
- BP contains an offset address in the current SS, which is used by instructions utilizing the based addressing mode.



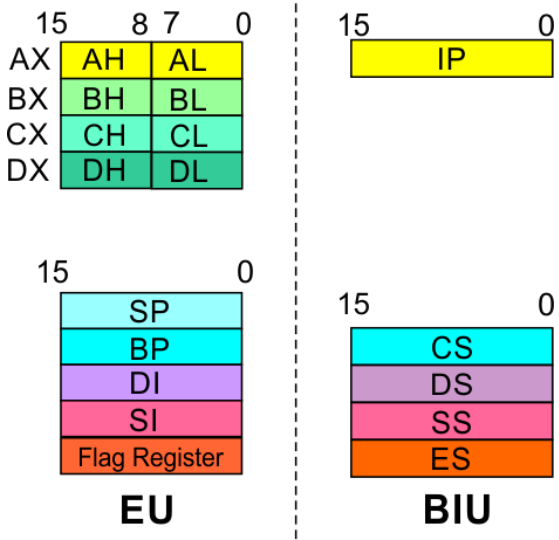
Stack pointer (SP)

- The stack is configured as a data structure that grows downward from high memory to low memory.
- At any given time, the SP holds the 16-bit address of the next free location in the stack.
- The stack acts like any other stack when there is a subroutine call or on an interrupt. ie. pushing the return address on a jump, and retrieving it after the operation is complete to come back to its original location.

EU Registers

Source Index (SI) and Destination Index (DI)

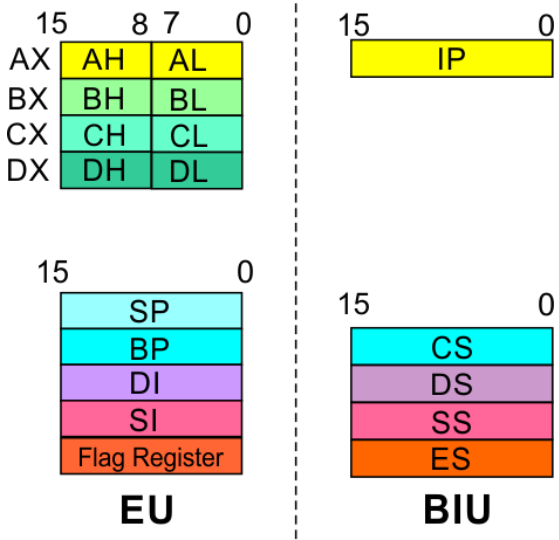
- **Used in indexed addressing.**
- **Instructions that process data strings use the SI and DI registers together with DS and ES respectively in order to distinguish between the source and destination addresses.**



EU Registers

Source Index (SI) and Destination Index (DI)

- **Used in indexed addressing.**
- **Instructions that process data strings use the SI and DI registers together with DS and ES respectively in order to distinguish between the source and destination addresses.**



EXECUTION UNIT – General Purpose Registers

Register	Purpose
AX	Word multiply, word divide, word I /O
AL	Byte multiply, byte divide, byte I/O, decimal arithmetic
AH	Byte multiply, byte divide
BX	Store address information
CX	String operation, loops
CL	Variable shift and rotate
DX	Word multiply, word divide, indirect I/O (Used to hold I/O address during I/O instructions. If the result is more than 16-bits, the lower order 16-bits are stored in accumulator and higher order 16-bits are stored in DX register)

Pointer And Index Registers

- used to keep offset addresses.
- Used in various forms of memory addressing.
- In the case of SP and BP the default reference to form a physical address is the Stack Segment (SS- will be discussed under the BIU)
- The index registers (SI & DI) and the BX generally default to the Data segment register (DS).

SP: Stack pointer

– Used with SS to access the stack segment

BP: Base Pointer

– Primarily used to access data on the stack

– Can be used to access data in other segments

- SI: Source Index register
 - – is required for some string operations
 - – When string operations are performed, the SI register points to memory locations in the data segment which is addressed by the DS register. Thus, SI is associated with the DS in string operations.
- DI: Destination Index register
 - – is also required for some string operations.
 - – When string operations are performed, the DI register points to memory locations in the data segment which is addressed by the ES register. Thus, DI is associated with the ES in string operations.
- The SI and the DI registers may also be used to access data stored in arrays

Architecture

Execution Unit (EU)

Flag Register

Auxiliary Carry Flag

This is set, if there is a carry from the lowest nibble, i.e, bit three during addition, or borrow for the lowest nibble, i.e, bit three, during subtraction.

Carry Flag

This flag is set, when there is a carry out of MSB in case of addition or a borrow in case of subtraction.

Sign Flag

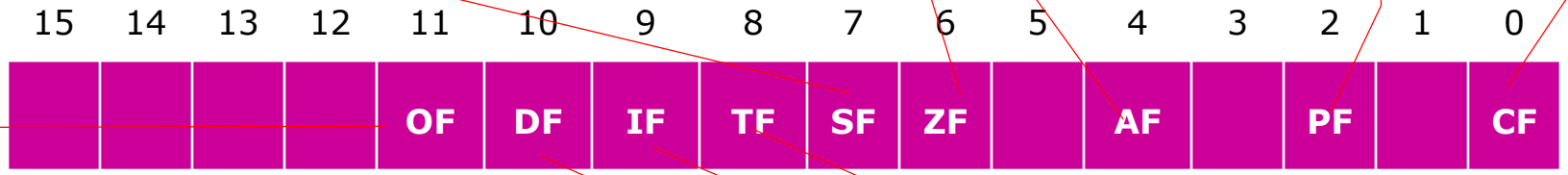
This flag is set, when the result of any computation is negative

Zero Flag

This flag is set, if the result of the computation or comparison performed by an instruction is zero

Parity Flag

This flag is set to 1, if the lower byte of the result contains even number of 1's ; for odd number of 1's set to zero.



Over flow Flag

This flag is set, if an overflow occurs, i.e, if the result of a signed operation is large enough to accommodate in a destination register. The result is of more than 7-bits in size in case of 8-bit signed operation and more than 15-bits in size in case of 16-bit sign operations, then the overflow will be set.

Tarp Flag

If this flag is set, the processor enters the single step execution mode by generating internal interrupts after the execution of each instruction

Direction Flag

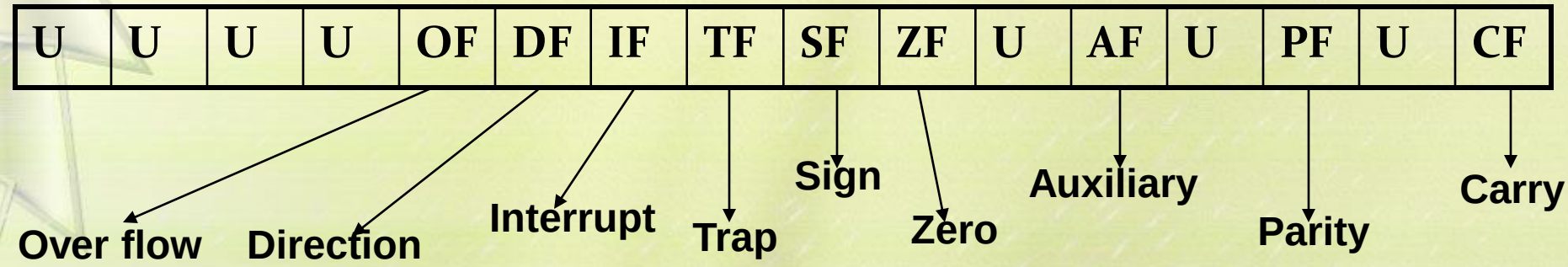
This is used by string manipulation instructions. If this flag bit is '0', the string is processed beginning from the lowest address to the highest address, i.e., auto incrementing mode. Otherwise, the string is processed from the highest address towards the lowest address, i.e., auto incrementing mode.

Interrupt Flag

Causes the 8086 to recognize external mask interrupts; clearing IF disables these interrupts.

EXECUTION UNIT – Flag Register

- A flag is a flip flop which indicates some conditions produced by the execution of an instruction or controls certain operations of the EU .
- In 8086 The EU contains
 - a 16 bit flag register
 - 9 of the 16 are active flags and remaining 7 are undefined.
 - 6 flags indicates some conditions- status flags
 - 3 flags –control Flags



U - Unused

EXECUTION UNIT – Flag Register

Flag	Purpose
Carry (CF)	Holds the carry after addition or the borrow after subtraction. Also indicates some error conditions, as dictated by some programs and procedures .
Parity (PF)	PF=0;odd parity, PF=1;even parity.
Auxiliary (AF)	Holds the carry (half – carry) after addition or borrow after subtraction between bit positions 3 and 4 of the result (for example, in BCD addition or subtraction.)
Zero (ZF)	Shows the result of the arithmetic or logic operation. Z=1; result is zero. Z=0; The result is 0
Sign (SF)	Holds the sign of the result after an arithmetic/logic instruction execution. S=1; negative, S=0

Flag	Purpose
Trap (TF)	A control flag. Enables the trapping through an on-chip debugging feature.
Interrupt (IF)	A control flag. Controls the operation of the INTR (interrupt request) I=0; INTR pin disabled. I=1; INTR pin enabled.
Direction (DF)	A control flag. It selects either the increment or decrement mode for DI and /or SI registers during the string instructions.
Overflow (OF)	Overflow occurs when signed numbers are added or subtracted. An overflow indicates the result has exceeded the capacity of the Machine

Execution unit – Flag Register

- Six of the flags are status indicators reflecting properties of the last arithmetic or logical instruction.
- For example, if register AL = 7Fh and the instruction ADD AL,1 is executed then the following happen

AL = 80h

CF = 0; there is no carry out of bit 7

PF = 0; 80h has an odd number of ones

AF = 1; there is a carry out of bit 3 into bit 4

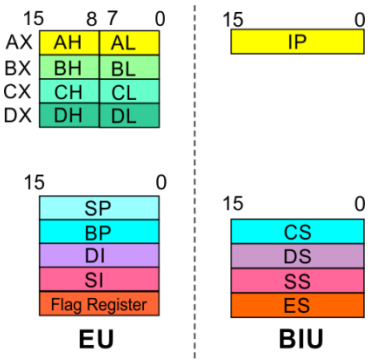
ZF = 0; the result is not zero

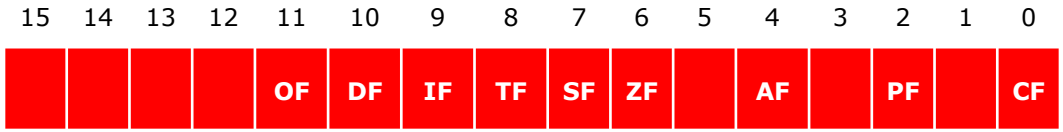
SF = 1; bit seven is one

OF = 1; the sign bit has changed

Architecture

8086 registers categorized into 4 groups





Sl.No.	Type	Register width	Name of register
1	General purpose register	16 bit	AX, BX, CX, DX
		8 bit	AL, AH, BL, BH, CL, CH, DL, DH
2	Pointer register	16 bit	SP, BP
3	Index register	16 bit	SI, DI
4	Instruction Pointer	16 bit	IP
5	Segment register	16 bit	CS, DS, SS, ES
6	Flag (PSW)	16 bit	Flag register

Architecture

Registers and Special Functions

Register	Name of the Register	Special Function
AX	16-bit Accumulator	Stores the 16-bit results of arithmetic and logic operations
AL	8-bit Accumulator	Stores the 8-bit results of arithmetic and logic operations
BX	Base register	Used to hold base value in base addressing mode to access memory data
CX	Count Register	Used to hold the count value in SHIFT, ROTATE and LOOP instructions
DX	Data Register	Used to hold data for multiplication and division operations
SP	Stack Pointer	Used to hold the offset address of top stack memory
BP	Base Pointer	Used to hold the base value in base addressing using SS register to access data from stack memory
SI	Source Index	Used to hold index value of source operand (data) for string instructions
DI	Data Index	Used to hold the index value of destination operand (data) for string operations

8086

- The 8086 is a 16-bit microprocessor chip designed by Intel, which gave rise to the x86 architecture; development work on the 8086 design started in the spring of 1976 and the chip was introduced to the market in the summer of 1978.
- The Intel 8088, released in 1979, was a slightly modified chip with an external 8-bit data bus (allowing the use of cheaper and fewer supporting logic chips and is notable as the processor used in the original IBM PC.

